NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A SYSTEM ARCHITECTURE AND MIGRATION PLAN FOR THE STUDENT SERVICES DEPARTMENT OF THE MARINE CORPS INSTITUTE

by

Clayton O. Evers, Jr.

September 1997

Thesis Advisor:	Magdi Kamel
Associate Advisor:	Mark Nissen

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 1997	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE A SYSTEM ARCHITECTURE AND MIGRATION PLAN FOR THE STUDENT SERVICES DEPARTMENT OF THE MARINE CORPS INSTITUTE	5. FUNDING NUMBERS
6. AUTHOR(S) Evers, Jr., Clayton O.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Director of the Marine Corps Institute, Washington Navy Yard, 912 Poor St. S.E., Washington, D.C. 20391-5680	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This thesis is part of a year long project that was undertaken by NPS students and faculty to develop a system architecture and migration plan for the transition from a legacy information system to a client/server based, open information system for the Marine Corps Institute (MCI). The primary objective of this thesis is to develop the technology architecture required to support the information systems of the Student Services Department (SSD) of MCI and to address the complex issues of system migration.

This thesis conducts an analysis of existing hardware and software, defines a technology architecture that will support the operational requirements of the data and business process model developed by other team members, and proposes a migration plan to transition from the current architecture to the proposed architecture that addresses both technical and human factor issues.

The thesis culminates in specific recommendations for MCI with regard to the hardware, software, and migration issues.

14. SUBJECT TERMS technology architecture, system migration, Enterprise Architecture Planning, Open Systems	15. NUMBER OF PAGES 151
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited.

A SYSTEM ARCHITECTURE AND MIGRATION PLAN FOR THE STUDENT SERVICES DEPARTMENT OF THE MARINE CORPS INSTITUTE

Clayton O. Evers, Jr.
Major, United States Marine Corps
B.A., University of Florida, 1984

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 1997

ABSTRACT

This thesis is part of a year long project that was undertaken by NPS students and faculty to develop a system architecture and migration plan for the transition from a legacy information system to a client/server based, open information system for the Marine Corps Institute (MCI). The primary objective of this thesis is to develop the technology architecture required to support the information systems of the Student Services Department (SSD) of MCI and to address the complex issues of system migration.

This thesis conducts an analysis of existing hardware and software, defines a technology architecture that will support the operational requirements of the data and business process model developed by other team members, and proposes a migration plan to transition from the current architecture to the proposed architecture that addresses both technical and human factor issues.

The thesis culminates in specific recommendations for MCI with regard to the hardware, software, and migration issues.

TABLE OF CONTENTS

ACKNOWLEDGMENT

The author would to acknowledge the financial support of the Marine Corps Institute in funding the travel and equipment purchased during the research phase of this thesis, the effort of Dr. Frank Barrett in assistance with Chapter VI, and the patience and understanding of his family for the support they provided during the creation of this thesis.

I. INTRODUCTION

Many organizations have initiated efforts to migrate from their current outdated legacy systems to open system architectures that support flexible business practices and take better advantage of emerging technologies. This research addresses one such effort by developing a technology architecture for a target hardware and software platform and a migration plan to transition from the legacy system to a contemporary system.

A. BACKGROUND

The Marine Corps Institute (MCI) was established to "develop, publish, distribute, and administer distance training and education materials to enhance, support, or develop required skills and knowledge of Marines and to satisfy other training and education requirements as identified by the Commanding General, MCCDC." MCI is the primary distance learning activity for the United States Marine Corps. The ability of MCI to perform this important mission has been significantly hindered by its outdated information system. The department most affected by the deficiencies of the current system is the Student Services Department (SSD).

In response to the system shortcomings, and in order to improve system flexibility for better student support, MCI initiated a project to redesign and modernize their information system using modern development techniques and an open hardware and software architecture. In addition, MCI is also reviewing and redesigning their business processes to better support its mission and to better keep pace with current advances in training and education.

B. OBJECTIVE

This research is part of a year long project whose overall objective is to migrate a legacy system of the Student Services Department (SSD) at MCI to a contemporary client/server open system.

The primary objective of this research is to develop a technology architecture to support the information systems of SSD and to address the complex technology and human resources issues of system migration. Fundamental principles guiding the formulation of the technology architecture are developed and used in the design; the "as is" architecture is documented and integrated with the "to-be" architecture wherever possible; and strategies, concerns, and considerations for the migration of the current system to the target system are developed.

C. RESEARCH QUESTIONS

This research addresses the following primary research questions:

1. Can a technology architecture be developed to support the current and future needs of the Student Services Department at the Marine Corps Institute?
2. Can existing hardware and software used by the Student Services Department of the Marine Corps Institute be successfully migrated to an open system architecture?

In addition, the following subsidiary questions must be addressed:

1. Can Enterprise Architecture Planning support all the necessary requirements of this transition?

2. What is the current state of the Marine Corps Institute Automated Information System (MCIAIS)?
3. What combinations of hardware and software should be used to meet new system requirements within the given fiscal limitations?

D. SCOPE

As indicated, this research is part of a large project aimed at the development of a comprehensive system architecture and migration plan for MCIAIS. As such, this thesis is closely coordinated with the work of four other students. These students are conducting research and preparing theses related to the development of the business processes in SSD; development of a data model to support the business processes; and development of a Graphical User Interface (GUI) and proof-of-concept prototype.

The focus of this research is on documentation of the current hardware and software environment, the design of a technology architecture to support future hardware and software requirements, and development of a migration plan to transition from the current system to the future system.

The scope of the technology architecture and migration plan presented in this research is limited to the systems in use by SSD, and does not address in detail the system architecture requirements or design of the enterprise system for MCI.

E. METHODOLOGY

Two distinct methodologies were used in the development of this thesis. The first was used for the development of the technology architecture, and the second was used for the development of the migration plan.

1. The Technology Architecture

The first methodology used in this thesis is based on Steven Spewak's Enterprise Architecture Planning model, specifically addressing tasks, timelines, issues for consideration, and the philosophical approach of the definition of the technology architecture.

2. The Migration Plan

The second methodology used in this thesis is the framework of system migration developed by Brodie and Stonebraker. This methodology is referred to as the "Chicken Little Approach," and is one of the two proposed methods of system migration based on actual migration efforts that are documented by Brodie and Stonebraker. The "Chicken Little Approach" provides an incremental method for implementing a contemporary system, and therefore best suits the MCI environment.

Background information and specific details regarding the existing architecture were collected through interviews with key MCI staff. Capabilities and requirements to support open system architectures were obtained from literature, World Wide Web, and personal interviews with NPS staff. System requirements were obtained from interaction with other students involved in the MCI project and responsible for the development of the data and process models. The primary limitation was interaction with the client, and obtaining required feedback in a timely fashion, due largely to competing priorities and geographic separation.

F. ORGANIZATION OF STUDY

This thesis is organized to address the objectives of the study in six chapters. Chapter II discusses the Enterprise Architecture Planning (EAP) methodology in detail, describing the four layers and seven components of the methodology. The chapter concludes with a discussion of some of the issues of concern for planners adopting this methodology.

Chapter III provides an overview of the current system, addressing its hardware and software environment, peripherals, networking environment and applications. It then provides an overview of the target client/server system.

Chapter IV describes the actual technology architecture for MCI as it was developed under the EAP methodology. It proposes three technology platforms options for consideration, and analyzes the advantages and disadvantages of each.

Chapter V presents the migration plan for this study. It provides an overview of migration planning, a discussion of the strategies and constraints of system migration, a detailed migration strategy for MCI, and recommendations concerning migration.

Chapter VI is an exploration of the human factors affecting this transition effort. The transition effort is analyzed using a human systems framework, and the issues of change management for this project are explored in detail with recommendations for improved transition management.

Chapter VII contains the conclusions and recommendations for this study. Appendix A contains an abbreviated Information Resource Catalog (IRC) that was used in the development of the technology architecture, and Appendix B is a system map, which depicts the connectivity of the entire information system at MCI.

II. ENTERPRISE ARCHITECTURE PLANNING METHODOLOGY

The purpose of this chapter is to provide the reader with a broad overview of the methodology used in developing the proposed architecture definition for the replacement information system for Marine Corps Institute (MCI). The methodology is first discussed as it applies to the enterprise level, then specifically as it applies to the development of the technology architecture. The methodology outlined in this chapter is adapted from the EAP methodology by Steven Spewak [Ref. 1].

A. ENTERPRISE ARCHITECTURE PLANNING

The greatest difficulty for information technology (IT) specialists in supporting the business needs of today's organizations is the confusion and incoherence of the planning process [Ref. 1]. Enterprise architecture planning (EAP) is a methodical attempt to provide a structured framework for the conduct of effective IT planning.

1. Definition and Components

Spewak defines enterprise architecture planning as "the process of defining architectures for the use of information in support of the business and the plan for implementing those architectures [Ref. 1]." The plan is to cover the three basic types of architectures: data, applications, and technology architectures. The important distinction drawn by this methodology is that of definition and design. Definition of a system is answering the question of "what" for a new system, design of a system answers the question of "how". EAP is used for the process of architecture definition. After the architecture has been properly defined, other processes can be used to design and implement the architecture.

EAP must combine the definition of the required architecture with a supporting plan that details when the architecture will be implemented. Without supporting implementation plans, the EAP team should keep in mind that the end product must go beyond simple definition and include in the final report a supporting plan for implementation.

2. The Four Layer Model

Spewak defines seven components of EAP, grouped in four layers to successfully complete the plan, see Figure 1 [Ref. 1]. His first layer is Planning Initiation, which emphasizes that the EAP must be started correctly, with the right tools, right people, and the right expectations. The second layer components are business modeling or a full analysis and understanding of the current business practices and the information that support them.

The third layer is the heart of EAP. The components of this layer are the definitions of the data, application and technology architectures. The data architecture is defined first, then the application architecture is defined around the data, then the technology architecture is developed last from the data and application architecture. While it may seem quicker or easier to subdivide these architecture developments among different teams, or to develop them in parallel, Spewak warns against it, stating "this approach does not work [Ref. 1]."

The fourth layer is the Implementation/Migration plan component. This component defines the sequence and schedule for implementation and the migration path, as well as some cost benefit analysis. This is the plan for how to get to the desired end state for the proposed system.

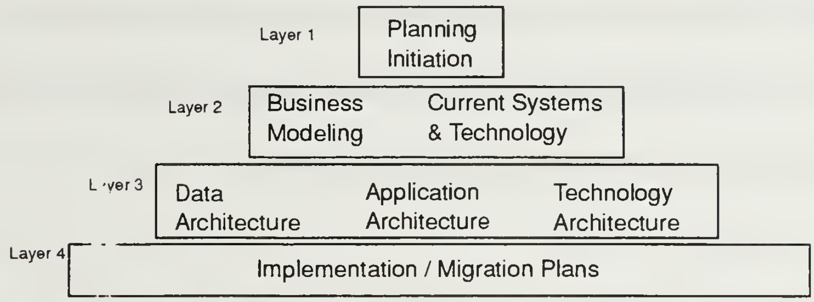


Figure 1. The Layers of EAP.

B. THE COMPONENTS OF EAP

As described above, the four layers of the methodology are broken down into seven components. This section is intended to provide an overview of each of the components of the methodology.

1. Planning Initiation

There are seven major steps for the conduct of planning under the EAP methodology. This section will briefly introduce those steps, formulated by Spewak [Ref. 1]. By following these steps, the planner can begin an EAP effort that is properly scoped, scheduled adequately, and undertaken with the appropriate personnel.

a. Determine Scope and Objectives

The first step in defining the scope is to actually define the enterprise. Since the goal of EAP is to allow the entire organization to share its data, the planner must ensure that all elements of the enterprise that have a need for data are identified. The scope must be defined to include all elements of the organization that will need to share data, especially across organizational boundaries. The planner must resist the temptation to narrowly define the scope of the EAP along departmental lines, because a department will almost certainly need to share data with other elements of the enterprise. An EAP scope that is too narrow will result in a product that is inflexible, difficult to integrate, and which falls short of the expectations held for a new system.

b. Create a Vision

This step is a difficult one for even the best information systems (IS) planners. The EAP group must make a concerted effort to understand the vision and goals of the business, to determine IS goals that will support the goals and objectives of the core business functions. Depending on the role of IS leadership, they might have been included in the development of those business goals; if not, the IS leadership must actively seek the stated and unstated vision and goals from business reports and documents.

c. Adapt a Methodology

This step is where the methodology is tailored to the specific needs of the organization. Whatever planning structure is being adapted for EAP use, it must remain flexible and adaptable, use automated tools, be compatible with the culture and politics of the organization, update the organization constantly, and result in a long-range implementation plan. The EAP methodology can be based on techniques for planning already in use at the organization, developed from scratch internally, or developed with the aid of outside specialists. As long as it can be adapted to the peculiar needs of the organization, it will still support the long range planning goals.

d. Arrange for Resources

The IS leadership must prepare for the EAP effort by dedicating the appropriate computer resources and other product resources that might be necessary for the EAP team to successfully complete the project. This might include data access and reports, administrative support, programming support and access to networks and mainframes used by the organization.

e. Assemble the Planning Team

Spewak identifies this step as the most critical. The team leader must possess strong leadership skills and be fluent in the methodology as well as the core functions of the business. The team itself must be educated in EAP, understand the process, and be committed to its success. They must have credible reputations in the organization, and must be willing and able to work together on this challenging task.

f. Prepare the EAP Workplan

The EAP workplan is the project management tool for the EAP effort. It outlines the timeline for project completion with milestones to keep the project on track, along with a plan for all team activities. The time schedule is critical for a plan of this scope because if the effort falls behind it will likely fail. Failure to produce anticipated deliverables may diminish credibility with management and endanger the project as well.

g. Obtain Management Approval

Like any other project consuming resources, the EAP initial plan will have to be presented to management for approval. This can be done throughout the initiation phase, or once the plan for the effort is completed. The group leader must prepare executive level briefs, detail the plan, listen to their questions, concerns and comments, and be prepared to address those issues. After approval is obtained, the decision should be widely published throughout the organization, and general EAP familiarization training should begin for the entire enterprise.

2. Business Modeling

Business modeling is done in order to make sure the business is properly defined. The model is the mechanism that will be used by the designers and users to ensure that all the processes of the organization are being considered for the new system. Spewak defines the purpose of the business model as "to provide a complete, comprehensive, consistent knowledge base that can be used to define the architectures and implementation plans [Ref. 1]." He divides the business modeling into two parts: A preliminary business model, and a complete business model.

a. Preliminary Business Model

The preliminary business model is composed of three steps. The first step is to document the organizational structure of the enterprise. This knowledge of the structure will help the EAP team decide whom to interview, and will also help in determining data sharing requirements and application sharing requirements later in the system development. The information may be available immediately to the team, but the team should validate the organizational information to ensure no undocumented changes have occurred.

This documentation will identify the business locations of the organization, and relate them to the organizational units. Business locations must be independently evaluated to determine if they are an attribute of organizational units or a separate data structure in the EAP database. This step must also document the business goals and objectives. These goals, objectives and critical success factors should be ranked in order of importance. Annual budgets and organizational plans should also be included here to contribute to sequencing of applications in accordance with organizational importance.

Finally the output of the first step is the production of the organizational units, reporting structure, locations, and business goals. These data structures and reports must be reviewed by the team and by the business units for accuracy in documenting relationships for the new system design.

The second step is identification and definition of functions. Identification of the functions of a business is the same as identifying the business. Because proper function definition is so critical to application development, this step is probably the most important in the entire methodology. The quality of the system architecture will be largely based on the quality of the business model, which is derived from the definition of the business functions.

The third step is the distribution of the business model. This includes the production of the output of the business modeling process, the physical distribution of that output, and the feedback from the organization regarding the accuracy of the model. Management will receive an updated presentation at the conclusion of this project.

b. Complete Business Model

The chief difference between the complete business model and the preliminary business model is the interview process. The complete business model will include the result of interviews conducted after the organizational analysis has been completed to decide who should be interviewed, what the right questions are, and how the information obtained from the interview process affects the business model. When this analysis is complete, so is the business model.

3. Current Systems and Technology

The corps of the current systems and technology component is the Information Resource Catalog (IRC). The IRC is the summary listing of all the components of the entire information system. It is not a data dictionary or an equipment inventory, but rather a broad scope picture of the major elements of the information system.

The IRC will show the distribution of major computer resources throughout the enterprise, and defines the architecture currently in place. Collection of the data necessary to complete the IRC is not a trivial task. It requires considerable time and effort and must be completed before the architectural phases of EAP are begun. The IRC for this study is included as Appendix A.

4. Data Architecture

The data architecture is the component of definition for the elements of data needed to support the business functions identified in the business model. Like the other components of EAP the definition of the data architecture is divided into steps. The first step is to list the candidate data entities for definition. This is usually a short step, and can be completed in a few brainstorming sessions. The candidate lists should include some preliminary attempts at definition, and indications as to the functions and use of the entities.

The second step is to define the data entities, their attributes, and the relationships between the entities. There are several methodologies available for this process. Spewak recommends the Entity-Relationship diagram method. Because of requirements for standard data modeling in the Department of Defense, the IDEF1X methodology was used to model the data architecture for MCI.

The third step is to relate the data entities to the business model developed earlier. In this step the team will determine which data entities are "created, retrieved, updated, and deleted by business functions" [Ref. 1]. The primary tool for this association of entities with business functions is the CRUD matrix. By the use of this matrix, the team can determine which entities are created, reference, updated or deleted by business functions.

The fourth step is distributing the data architecture. Like the business model, this involves the collection of all the information generated during this phase, development of the data model itself, the preparation for presentation, and the actual presentation of the data model and architecture to the organization and to management. Feedback from the organization and from management should then be considered for incorporation into the data model.

5. Application Architecture

The purpose of the application architecture is "to define the major kinds of applications needed to manage the data and support the business functions of the enterprise" [Ref. 1]. This component is not a design for the system or a detailed requirements analysis for the system. Analogous to the data architecture, the first step of the application architecture is to identify all possible applications that might be required to manage the data and support the business functions of the organization. Current applications in use should be listed, as well as applications with potential for improving business processes.

The second step is to define all the candidate applications. Application definitions should describe what an application does, not how it does it. The application descriptions should not be linked to a particular technology, but described in general, non-technical terms.

The third step is linking the applications defined in step two to the functions of the business. Matrices are used to display the correlation of the applications to the supported business functions. If functions are discovered to have no application support, it must be determined why this situation exists.

The fourth step is to analyze the impact of the developing application architecture on the current applications. The application architecture can be compared to the IRC to determine which applications are completely replaced by new applications, which are partially replaced, and which applications will be retained, possibly with some enhancement.

The final step is the presentation and feedback step for the organization. The application architecture must be briefed to management and to the organization, and feedback of the architecture is considered for inclusion to the final product.

6. Technology Architecture

Spewak defines the purpose of the technology architecture as "defining the major kinds of technologies needed to provide an environment for the applications that are managing data" [Ref. 1]. The definition of the technology architecture is one of the specific goals of this study, and is discussed in Chapter IV. It is useful to introduce it here in order to provide the reader with a proper context for understanding the specifics of the technology architecture. There are four steps that make up Spewak's technology architecture framework.

The first step is to identify technology platforms and principles. This step establishes the guidelines for the entire architecture development. The principles that will govern the development of the technology architecture must be based on trends and directions of the IS industry. A wide variety of literature should be studied to ensure critical industry potentials at least receive due consideration. After the principles are defined, the team will compile a list of the potential technology platforms for consideration.

The second step is to define the technology platforms and distribution of those platforms. With the principles of development defined, the team can develop their strategy for the distribution of the applications and data. All business locations affected by the architecture will be identified by location and function. The physical and conceptual location of the data must also be determined in the distribution plan. Finally, a definition for the configuration of the technology platform must be developed. This conceptual architecture must address the conceptual workstation (user access), the conceptual enterprise network (input/output, storage and telecommunications) and the business systems architecture (implementing and maintaining applications and data of enterprise).

The third step is to relate the technology platforms to the applications and business functions developed in the earlier components. In the planning guidelines above, the importance of linking the EAP to core business functions was discussed. In this extension of that philosophy, the technology platform defined must be related to the business functions that will use them and to the applications architecture that requires that technology.

The fourth step in Spewak's technology architecture is to distribute the technology architecture. The documents defining the architecture must be prepared in a clear, useful format, then presented to executive management. The team must be prepared to discuss the potential gains and risks to the organization, data integrity and security issues, and implementation concerns. The briefing may raise new issues for the team, and those issues must be considered for possible revision of the implementation plan.

7. Implementation/Migration Plans

The implementation plan is last of the components, and represents the bottom layer of the EAP methodology. Without a plan for implementation, the entire EAP effort will be accepted and shelved without further consideration. This plan will incorporate several project management techniques, and is a long term plan for the implementation of the EAP findings.

The first step in the implementation plan is a sequencing of the applications. Although it sounds like common sense, applications that create data should be implemented before applications that use data. The sequence of implementation should be data driven and adjusted to support the needs of the business.

The second step is to estimate the effort and resources required for implementation, and to produce a schedule. This step includes the acquisition estimates for software and hardware, the estimation of available resources to support the plan, and the use of project management techniques to produce a workable schedule for implementation.

The third step is a cost and benefit analysis of the plan. Financial concerns will be the driving force behind approval for implementation of the plan. Executive decision makers must have solid data on economic benefits, rates of return and congruence with long term business goals in order to provide the financial backing for an effort as costly as a system overhaul.

The fourth step is to determine the critical success factors for the plan, and to prepare recommendations on how to satisfy them. These critical success factors may include such things as new organizations in the enterprise, new development methods, significant budgetary changes, training for personnel, and reorganization of the IS function. All of these factors will involve change, and the transition management must also be considered. The factors influencing transition management will be discussed in detail in Chapter VI.

C. ISSUES REGARDING EAP

Spewak addresses several additional issues for consideration by planners considering the selection of a planning methodology. He contrasts EAP with traditional planning, and provides an overview to some of the more common obstacles faced by planners using EAP.

1. EAP Versus Traditional IS Planning

Spewak summarizes the difference between EAP and traditional IS planning as a difference of the driving factors. He maintains that traditional IS planing is driven by process and technology. His EAP methodology is driven by data and business. This fundamental element of focus is what makes EAP different.

EAP seeks to first understand the business. The planning team conducts the analysis to fully understand what the business does and what information is required to do those things. By understanding the function of the business first, the architecture definition is driven by the needs of the business, not the technology the business is currently using.

The step associated with data development is a complete reversal from the norm. EAP defines the data before the applications. The component of data architecture defines all the data needed to meet the business needs first, then defines the applications required to manage that data in the application architecture.

Another reversal under EAP is the implementation plan. Traditional implementation strategy would be to implement at the highest level of visibility. EAP maintains that data must be created before it can be used, so the first implementations should be at the creation level, which is usually lower visibility than the management level.

The last major difference between EAP and traditional IS planning is the intended focus. Traditional IS planning has focused on short term solutions to immediate problems.

The goal of EAP is a long term solution with enough flexibility to adapt to any problem that may arise from the business, not just those areas with high profitability or immediate payback.

2. The Zachman Framework

The EAP methodology has its basis in the Zachman framework, which was first published in 1987 [Ref. 2]. This was the first paper to identify a "framework" for analysis of information systems. Zachman defined six levels from which a system could be viewed. It was also the first definition of the three layers underlying architectural planning: data, process, and network. EAP is a refinement of the Zachman framework, and focuses on the top two layers -- the ballpark view and the owner's view. This is the basis for EAP developing the definition, but not the design of information systems.

3. Obstacles to EAP

Spewak identifies several common obstacles that planners should be prepared to overcome in the course of EAP efforts. These obstacles range in levels of importance, and some of the more crucial obstacles are summarized here.

a. Top Management

Support of top management is the key to any successful change effort, and EAP is no exception. The key decision makers must be educated about EAP, and realize that the EAP effort is key to reaching their business goals.

b. Resources

Many of the biggest obstacles to effectively implementing an EAP involve the lack of proper financial resources. EAP is a significant undertaking for the organization; as such it will involve the dedication of personnel and financial resources consummate with an effort of its scope. The proper dedication of financial resources can be an effective measure of the true support of top management. Failure to assign the required amount of qualified personnel to an EAP effort is a sure sign that top management is not fully supporting the plan.

The priority of resource allocation is also another indicator of potential success for EAP. There are likely numerous IS projects waiting in a backlog in any IS organization. If EAP is to be effective, it must be conducted before things that have been waiting for the front of the queue. The impact of a new system design should be able to incorporate many of the requirements of backlogged requests if they prove to be truly useful in support of the business.

The other major resource question that EAP planners will face is that of rates of return on investment. The proper conduct of EAP will certainly have a substantial up-front cost. There will not be an immediate return on EAP efforts, and the backers of EAP must ensure that the plan is given support to be able to get to the design phase of development, or there will never be a return on the investment, intrinsic or otherwise. The biggest danger of this type of failure will be the political impact it will have, which is discussed below.

c. Politics

The first political problem to overcome is an extension of the argument above. The IS department of any organization has probably been responsible for projects that were substantially late and over budget. If the IS department is reflective of the IS field as a whole, perhaps all of the department projects have been substantially late and over budget. This legacy will be the first hurdle in selling new methodology requiring the dedication of financial resources and top people.

Top management will certainly factor the past performance of the IS department into the support it gives the EAP effort. In order to successfully define and design the organization, the plan must be formulated by people from various places across the enterprise. Planners must ensure the people involved in EAP carry strong credibility in the organization, so these people must be educated in EAP techniques first.

The education process itself is wrought with political issues that must be addressed by the EAP proponents. Technologically oriented people are even more likely to see EAP as a "flavor-of-the-month" and dismiss it without due consideration. IS leaders must find the proper motivation and education tools to ensure buy in by the key people in the IS department itself.

There are many political facets to embarking on any new way of doing things. The last area to discuss here is again internal to the IS department. The responsibilities for many EAP functions will cross normal organizational boundaries. There will be a tendency to try the "divide and conquer" approach to this effort. The EAP planner must resist this, because shared responsibility for EAP will significantly reduce its likelihood for success. The shared responsibility will blur authority lines and increase the political role of the human dynamics. Again, detailed discussion of the political factors affecting transition is found in Chapter VI.

d. Corporate Culture

Since EAP will require organized effort across the enterprise, there will be interaction with sections outside the IS department that might not normally have that level of interface with IS planners. This integration of people is crucial to EAP's success. IS planners must understand the core business, and the goals for successfully reaching those core business strategies. Many IS departments possess an institutional arrogance regarding their role in the business. This cannot be allowed to enter into the EAP process, as it will poison the effort from the beginning.

e. Inexperience with EAP

Because EAP is a fairly new methodology, it is likely that many key IS personnel have never used it before. It is critical that the participants understand the methodology before attempting to initiate the planning, or the credibility of the entire effort will be at risk. An aggressive education program must be conducted to minimize this risk. This can be done through seminars, outside classes, or through the use of training consultants.

III. OVERVIEW OF CURRENT AND TARGET SYSTEMS

This chapter is intended to provide background information on the research sponsor and its current primary information system. It begins with a brief history of the Marine Corps Institute (MCI), and the development of their automated information system. It then examines in detail the composition of the existing information system and provides an introduction to the expectations of a replacement system.

A. HISTORY OF THE MARINE CORPS INSTITUTE

The Marine Corps Institute was established to "develop, publish, distribute, and administer distance training and education materials to enhance, support, or develop required skills and knowledge of Marines and to satisfy other training and education requirements as identified by the Commanding General, MCCDC." To accomplish its mission, MCI is organized into seven functional departments: education and operations, student services, information management systems, occupation specialty, professional military education, production, and logistics.

The mission of the Student Services Department (SSD) is to support the enrollment, grading and management of the Marine Corps distance education and training programs. It is the focal point of most past complaints from students, and has received the majority of attention as the logical place to begin to seek improvement in customer support. In support of its mission, SSD employs an automated information system (AIS) to automate the actions required to administratively support a student in one of the MCI correspondence programs, maintain student records, and produce necessary management reports.

B. HISTORY OF THE MARINE CORPS INSTITUTE AUTOMATED INFORMATION SYSTEM

The automated system, known as the Marine Corps Institute Automated Information System (MCIAIS) is a legacy system developed in the late 1970's. It runs on a Hewlett-Packard 3000/957 mini computer running the Hewlett-Packard MPE/iX operating system. MCIAIS is written in a Hewlett-Packard proprietary language called "Transact" and accesses a TurboImage hierarchical database.

MCIAIS has been modified without documentation for many years. As a result of these modifications, it no longer efficiently supports business requirements, and does not have the flexibility to support emerging business requirements generated by new technology development. As is typical of many legacy systems, MCIAIS suffers from many shortcomings:

- It has over 110 "spaghetti coded" programs that are difficult to maintain, modify, and upgrade.
- It does not have underlying data or process models
- Programs have poor functionality: Poor checks and balances; no statistical analysis capability; and limited "ad hoc" query capability.
- It utilizes a "closed" non-relational database.
- It does not support Graphical User Interfaces (GUI).

In response to these shortcomings, and in order to improve flexibility for better student support, MCI initiated a project to redesign and replace MCIAIS using an open hardware and software architecture. In addition, MCI is also reviewing and redesigning business processes to better support its mission and current advances in training and education. This effort is a recognition of the fact that the role of MCI in the Marine Corps has changed significantly in the last year. Because of directives from the Commandant of the Marine Corps, courses developed and administered by MCI have been directly linked to individual Marines' careers as a prerequisite to promotion. The impact of this linkage has been to significantly raise the visibility of the student support provided by MCI. The intent of this research is to improve that support through the definition of a new information system.

As discussed in Chapter II, the technology architecture development methodology was applied to this requirement to develop an architecture definition that would satisfy the business requirements of MCI. While the specifics of developing a technology architecture will be discussed in Chapter IV, the first step of the methodology requires an examination of the system currently in place, and is conducted in the remainder of this chapter.

C. CURRENT SYSTEM

In order to properly plan the migration from the legacy system to the target system, the current system must be described in detail, to determine which, if any, part of the current system will be useful in the target system. The current system was inventoried according to the Information Resource Catalog technique (see Appendix A) as detailed in Spewak's book [Ref. 1].

1. Current Hardware Environment

The current hardware environment is a composition of computers of various types and their supporting peripherals.

a. *Computers*

The center of MCIAIS is the Hewlett-Packard 3000/957 SX minicomputer with 160 MB of memory. It has been recently upgraded, and adequately supports the information system in its current form. There are currently three microcomputer servers located at MCI.

- Two Dell 486-66 MHz machines with 32 MB memory and 7.5 GB disk arrays
- Dell Pentium-60 MHz with 64 MB memory and a 8 GB disk array
- Automated Voice Response (AVR) System, Pentium with 32MB of memory and 1.2 GB storage for automated dialup customer support.

The other microcomputers in use at MCI are serving as either networked or stand alone personal computers. At the beginning of his study, there was a wide range of microcomputers with different capabilities. Many of the customer support personnel were using 386 based machines. Part of the effort to improve customer support focused on upgrading the older microcomputers to state-of-the-art computers. Currently all PC's at MCI are being replaced, or have already been replaced with Pentium 90 MHz microcomputers with at least 16MB of memory, some ranging up to 40MB of memory. All of the Pentium microcomputers have a minimum of 1.2 GB of storage.

b. *Peripherals*

The peripherals found at MCI in support of the various computing requirements include:

- One kodak digital camera
- Two flatbed scanners
- Thirty five laserjet printers

- One HP 1600 line printer
- One Xerox 4850 laser printer
- One compact disk storage array with six CD drives
- One rewritable CD drive for information storage
- One HP 6250 Streaming Tape drive
- One NCS model OP7-35 form input scanner for test grading
- One HP 6000 SCSI Storage system consisting of four 2GB disk drives
- Two 2GB SE SCSI disk drives
- One 2GB DDS DAT drive.

2. Current Software Environment

The current software environment covers a wide range of system and application software to meet the requirements of the users at MCI. There are several operating systems, at least two network operating systems and a host of applications that must be supported.

a. Operating System

The operating system running the HP minicomputer is a HP proprietary operating system known as MPE/iX version 5.5. It is a POSIX compliant operating system, familiar to the senior information systems (IS) staff, but novel to the junior members of the IS Staff. It is not accepted as a standard operating system in the Marine Corps, but has been widely used for financial applications in other Department of the Navy sites. It is flexible and open enough to support current system requirements, and even near term future requirements, but staff training in MPE/iX is a critical concern for future support.

The majority of the microcomputers at MCI are running Windows 95, but some of the personnel at MCI have yet to upgrade Windows 3.11 to 32 bit systems. All of the PCs in use at MCI are operating on a DOS 6.22 base. There is currently one microcomputer running Windows NT, version 4.0. This computer is designated as the Web server for MCI.

b. Network Operating System

The network operating system (NOS) currently used at MCI is primarily Banyan Vines version 6.3(0). Banyan Vines is currently the standard NOS for the Marine Corps. Banyan Vines is robust and flexible enough to support all the current needs of MCI, and personnel are well trained in its use.

c. Application Software

As typical of any organization, there are a wide variety of applications supported by the IS staff. Many of the applications support other business functions of MCI in addition to SSD. The categories of applications found at MCI are application suites, graphics, message preparation, project management, utility, e-mail and some specialty applications. A detailed listing of these applications can be found in the Information Resource Catalog located in Appendix A.

3. Current Networking Environment

The networking environment at MCI is the least flexible and most saturated portion of the information system. The IS staff has done a credible job supporting requirements with available resources, but the current network environment no longer supports the requirements of MCI, and is not likely to support the future requirements of an expanded information system and increased dependence on the information provided by that system. While the Local Area Network (LAN) is adequate for today's needs, the Wide Area Network (WAN) or outside connectivity is seriously inadequate to meet even current requirements.

a. Data Links

The current primary data connectivity outside of MCI is a 56 kbps (thousand bits per second) line to Headquarters, U.S. Marine Corps (HQMC). This bandwidth is totally inadequate, and is planned for upgrade to T1 immediately. This T1 will be attached to a Cisco 4500M router. In addition to this primary data link, MCI has he following data equipment:

- Five dial-up phone links, operating at 28.8 kbps
- Two Cabletron MMAC-8 Hubs with one management module (w/SNMP) per hub
- Four 24-port 10baseT ethernet modules for user ports
- One FDDI module per hub

To increase data throughput for MCI, a 24-port 100 mbps (million bits per second) switched fast ethernet hub is planned for application server traffic and high-speed connections for data processing personnel.

Data connectivity between MCI and the headquarters at Marine Barracks, 8th and I streets is provided by an AirLan/Bridge, with long range 11db directional antenna, providing a 2 mbs data transfer rate over spread-spectrum transmission in the 900 MHz frequency range.

b. Internet

All connectivity to the Internet is provided via the 56 kbps link to HQMC. With the role of the Internet expanding every day, the planned T1 upgrade to this link will prove insufficient for future growth as well. MCI has added an informational web site which is likely to increase the bandwidth demand on this T1 data link. While the role of the Internet in current business practice is not included in MCI's short term plans, the utility of an expanded Internet presence will certainly have to be a central element of focus for information system planning.

D. TARGET CLIENT/SERVER ENVIRONMENT

The most radical change in moving to a new information system for MCI will be the change from a mainframe/minicomputer or host centric environment, to a client/server environment emphasizing distributed computing. The level of distribution is determined by the planners based on the logical location of the required processing. The target environment must provide regulated access to shared resources by many clients at one time, with the information processing required by that access located where it best facilitates communication. The specifics of the target system requirements will be discussed in Chapter IV, but a general introduction is provided here to contrast the current MCIAIS.

1. Target Hardware Environment

The target hardware environment is one that provides state of the art processing power to both the client and server side of the system. It is designed for ease of installation of peripheral equipment, low mean time between failure rates, ease of maintenance, high scalability, disk mirroring for data redundancy, high security, and ease of operation for the IS personnel.

The target hardware can be minicomputer or a microcomputer based, as long as it supports open architecture and the need for increased flexibility and scalability. Specific alternatives for both types of target systems are discussed in detail in Chapter IV.

2. Target Software Environment

Open systems are the prevailing industry trend for operating systems. Any operating system chosen for the target system must conform to all open standards, and support the flexible addition of new equipment, ease of operability for IS personnel, high security, support for multiple protocols, high integrity, and reliable support. Open standards insure the ability to mix and match various protocols that might need to be integrated to support emerging business requirements.

Relational databases are the most prevalent database applications in industry today. The application software chosen for the target environment must support relational database structure to eliminate the limitations of the current hierarchical database.

3. Target Networking Environment

Like the target hardware and software environments, the target networking environment must emphasize flexibility. The key to flexibility in networking is bandwidth planning. MCI needs to begin planning for adopting an Internet strategy in the future business plan. Appropriation planning to obtain the required bandwidth to support this plan should begin now. The target networking environment should be planned around standard equipment for Marine Corps infrastructures to reduce training requirements for new personnel. Existing plans to upgrade to 100 mbps service to networked users is a step in the right direction, but this type of improvement must not be viewed as a final step, rather as an intermediate step in support of the ever increasing requirements of complex information system support.

IV. TECHNOLOGY ARCHITECTURE DEFINITION FOR MCI

This chapter further defines the technology architecture for MCI that was briefly described in Chapter III. It describes the principles of development, defines three technology platforms for consideration, associates the technology platforms with business functions and applications, and discusses the distribution of those applications and the supporting data.

A. ENTERPRISE TECHNOLOGY ARCHITECTURE

As Chapter II clearly pointed out, one of the critical tenets of EAP is that planning is conducted at the *enterprise* level. The scope of what could be considered an "enterprise" for this project was unclear as the following section illustrates.

1. Scope of the Enterprise Architecture

By industry standards, MCI is not considered a large enterprise. In fact, some definitions would not classify MCI as an enterprise at all. A case could be made that MCI is a single business unit, and the information system under development is to support the core business functions of that business unit. The business functions identified, however, are central to the mission of MCI and serve as the core business functions of the enterprise. Since the scope of the redesign effort for this study was narrowly defined around SSD, it was clear that a technology architecture could not be readily developed by the Naval Postgraduate School (NPS) project team for the entire enterprise. The development effort focused on the business functions of SSD, which represent the large majority of the core processes of MCI. Even though the scope of the study was intentionally set at the business unit, proper use of the EAP methodology required that the enterprise be given consideration in the development of the technology architecture. This meant that other processes outside of SSD that were identified by MCI personnel and team analysts as critical to the business processes of MCI must be added to the technology architecture under development.

The artificial scoping of the project created a difficult situation for the project team. Any recommendations made to MCI for potential technology platforms must have the potential and capability to support the requirements of the entire enterprise. The project team elected to add the key elements from the external business processes to increase the viability of the project.

While technology architecture recommendations for SSD are based on detailed analysis for the business processes and data requirements of SSD, the hardware and software chosen as the technology platform must be robust enough to support the immediate and eventual needs of the enterprise.

The primary business unit external to SSD with significant impact on the system was the Logistics Department. High level business functions and data entities for that department were modeled and data interfaces with SSD were identified. This information was considered in the development of the technology platforms applied in this chapter.

2. Principles of Development

The purpose of establishing principles of development for the enterprise is "to identify underlying principles for technology platforms and the potential platforms needed to support an enterprise wide, shared data environment" [Ref. 1]. This identification is the first step of the EAP technology architecture methodology. The following principles apply to the formulation of an enterprise technology architecture for the Marine Corps

Institute project [Ref. 1]:

- Client/server technology will be used for applications and database implementation.

- A common graphical user interface will be used by all applications.

- Data storage will use relational technology, and data access will employ SQL.

- Design will apply open system concepts, meaning operating systems should be:
  Portable: run across multiple vendor platforms;
  Scalable: run across a wide power range from small to large computers;
  Interoperable: run in a heterogeneous environment; and
  Compatible: preserve the investment in existing software and enable technology advances to be integrated with other components.

- System development methodology should employ object oriented techniques, information engineering methods, and be supported by CASE and repository tools from requirements analysis through code generation.

- Data should be captured once at its source.

- Data should be administered centrally and maintained for shared access.

- Information that is stored online will be continuously available.

- Distributed data and application systems will be implement where possible.

- The security of data, software, and hardware assets at all levels of the technology architecture will be maintained with security being as transparent as possible.

- Recoverability will be emphasized to ensure to protect the continuation of business by having:
  adequate and appropriate backups of all data;
  software with built-in error checking and recovery capabilities; and
  integration and compatibility of hardware with redundancies for critical operations

- Established Marine Corps standards for software will be followed.
- Data storage requirements must be compatible with World Wide Web access requirements

B. BUSINESS UNIT LEVEL TECHNOLOGY ARCHITECTURE

As indicated, only a minimal analysis was conducted for the development of a technology architecture for the entire MCI enterprise. The focus on the development of the technology architecture was for the business unit level, the Student Services Department.

1. Hardware Platforms

Three target hardware platforms were considered as suitable replacement alternatives for the existing system. All three hardware alternatives comply to a varying degree with the principles established in the previous section. Two of the hardware alternatives are minicomputer based while the third is microcomputer based. The requirements and considerations for the hardware were discussed in Chapter III.

2. Operating Systems

Each of the three hardware alternatives has its own operating system option. Three operating systems were considered. All three are considered to be relatively open systems. All three operating systems are flexible enough to adapt to the changing conditions expected in the near future, and all three operating systems run the selected database management software.

3. Application Development Environment

The primary application development tool recommended by the project team is Oracle® Developer 2000. This application development tool was chosen for its robust, industrial-strength development capability and its strong security features. Used in tandem with Oracle® Designer 2000, this suite is considered one of the leading application development tools in the industry. In addition to its favorable reputation, the Oracle® development suite is seamlessly integratable with Oracle® Server, the chosen database management system.

Other application development tools were considered and dismissed. Borland's Delphi, Cognos' Power Builder, and Microsoft's Visual Basic were all considered for application development, but none could match the advantages and fit of Oracle® Developer 2000. The project team, however, did not use the Designer 2000 development tool, because it does not support the IDEF modeling techniques, identified by the Department of Defense as the standard for data modeling. Consequently Er*win*® and its sister tool BP*win*® were chosen.

4. Database Management System

The database management system (DBMS) selected for the technology platform is Oracle® Server. Other DBMS were considered during research, but Oracle® was selected for two main considerations. It has been named as the standard large scale DBMS for the United States Marine Corps [Ref. 3], and training at no cost to MCI was available locally for the project team in Oracle® Server and Oracle® Development tools. In addition, NPS was installing the Oracle® Server products, and local expertise was available at the school.

C. DEFINITION AND EVALUATION OF PROPOSED BUSINESS UNIT LEVEL TECHNOLOGY ARCHITECTURES

This step is intended to look at the conceptual architecture at three levels: the conceptual workstation, the conceptual enterprise network, and the business systems architecture. The conceptual enterprise network is being considered because the technology architecture for the business unit level must also be capable of supporting enterprise requirements for MCI. This architectural definition is further subdivided into three options for consideration by MCI. Each of the three options is evaluated under the guidelines established for this step. In addition to EAP considerations, details affecting system migration are included for consideration with each option. Other details, such as communication specifics for the final system design, will necessarily be incorporated as the final design is developed by the implementation team. The intent here is to show a broad *conceptual* design without committing to exact specifications. In developing these options, a seven year system life was assumed and, in compliance with a MCI request, only initial costs were considered.

1. Option One - HP Minicomputer Server Running MPE/iX

a. Hardware/Software Considerations

This option keeps the existing HP 3000 model 957SX minicomputer server, while replacing the current database system and its applications with a relational database system and associated applications. The current minicomputer will require some upgrade/replacement of hardware and software to meet the expected growth requirements of MCI. The 957SX can be upgraded in incremental steps up to a model 987/200 which is six times more powerful than the existing minicomputer. There are four levels of upgrade available to MCI at increasing increments of cost. MCI has two microcomputer purchase initiatives currently underway to replace outdated computers. These efforts will satisfy client workstation requirements for SSD. Upgrades to these client workstations can be limited in scope to minor hardware upgrades if continued migration to a different platform is considered. This option is depicted in Figure 2.

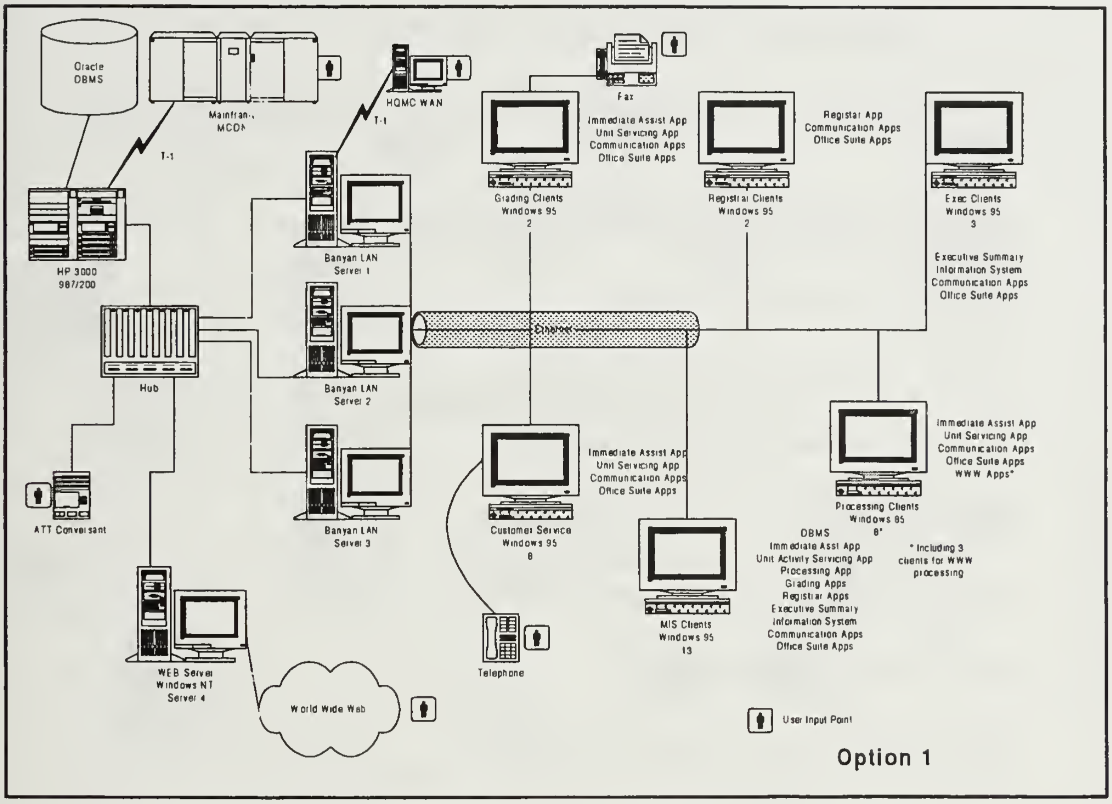


Figure 2. Option One Technology Architecture.

b. Migration Issues

The ability to use the existing hardware and operating system (MPE/iX) will enable an incremental transition for the MIS personnel. By adding the relational database to existing hardware and operating system, the MIS personnel can focus their training strategies on the transition to a new database management system without simultaneously having to learn new hardware and operating systems.

A main input data (MCTFS) for the new database is currently available on the HP 3000. This data can be imported easily into the new database without the purchase of any additional equipment. By implementing this option, the new database will be installed on the same server as the existing database, thus simplifying migration.

c. Advantages of Option One:

- Current personnel trained on HP 3000 hardware and MPE/iX operating system
- Support system already in place for HP products
- Potentially simplified migration
- Gateway product immediately available

d. Disadvantages of Option One:

- MPE/iX is not DOD or USMC standard
- MPE/iX is not a truly open system
- High yearly maintenance costs
- High upgrade costs to maximum capability
- Not responsive to future changes

2. Option Two - HP Minicomputer Server Running UNIX (HP/UX)

a. Hardware/Software Considerations

Option two requires a change to both the central hardware and operating system. The HP 3000 would be replaced with an HP 9000 model 969/200 running HP/UX, which is the HP version of the UNIX operating system. The HP 9000 969/200 is the next generation of HP minicomputer and is, according to benchmark tests, seven times more powerful than the current system.

Option two establishes a relational database on the new minicomputer, replacing the current database system and related applications. This database server will be capable of meeting the business needs of SSD for the required growth cycle (7 years). The two microcomputer purchase initiatives currently underway will satisfy client workstation requirements. This option is depicted in Figure 3.

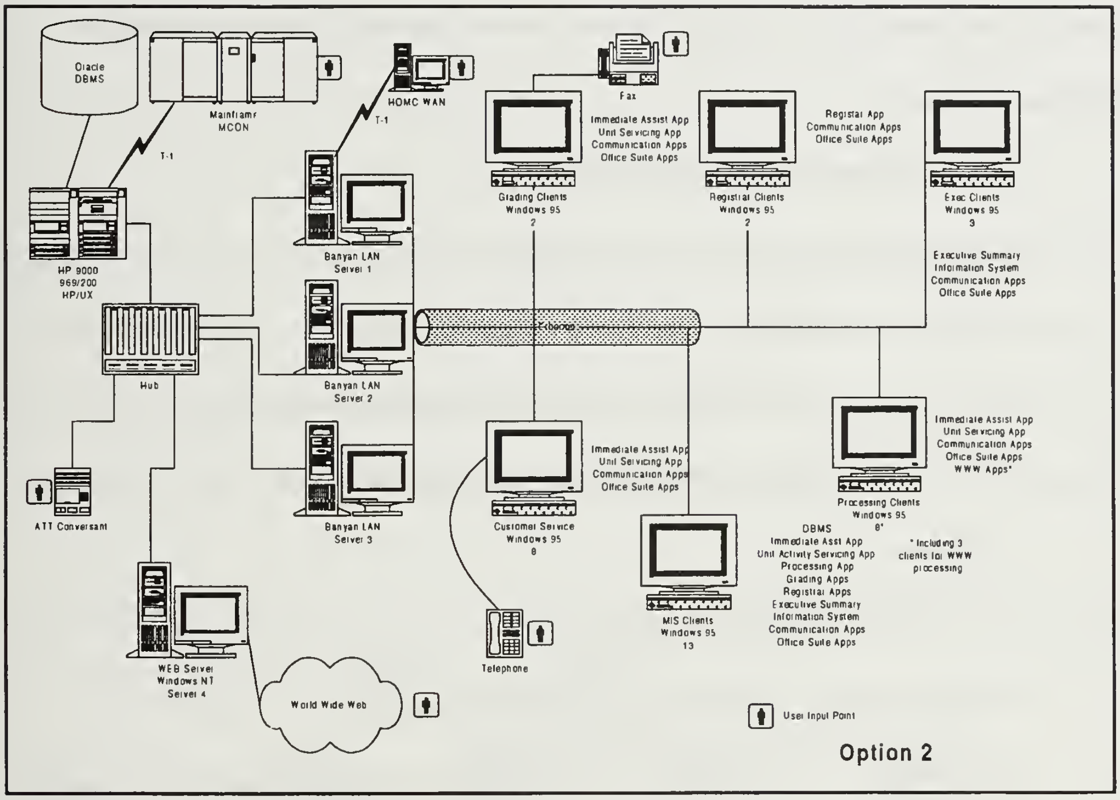


Figure 3. Option Two Technology Architecture.

b. Migration Issues

This option represents a greater level of change than option one, but less drastic change than option three. It will require extensive training of MIS personnel in UNIX, and this training requirement alone increases the complexity of the transition. Entry-level MIS personnel at MCI have had some UNIX exposure, but current training levels are inadequate for the requirements of this transition. UNIX is an open system, widely used in industry and DOD. Oracle® support and experience with UNIX platforms is the most developed. The likelihood of simplified interface with developing technologies for future applications and equipment is improved with the UNIX O/S. HP personnel will be available for hardware migration assistance to facilitate the cutover from one technology platform to the other. This solution is evaluated as having greater risk of failure than option one.

c. Advantages of Option Two:

- More open and more standard operating system
- UNIX O/S is preferred platform of Oracle®
- UNIX O/S provides better World Wide Web interface and design
- UNIX O/S simplifies external services (print gateway)
- Scalability

d. Disadvantages of Option Two:

- Significant learning curve for senior MIS personnel
- More complex migration due to the requirement of new hardware and software
- UNIX security weaknesses
- High yearly maintenance costs
- High investment may stymie future migration efforts

3. Option Three - Intel Based Server Running NT

a. Hardware/Software Considerations

This option represents a core shift in computing philosophy for MCI. It replaces the current minicomputer with a state-of-the-art multi-processor microcomputer.

It will take advantage of intensive industry development by Oracle®, and represents a significantly smaller capital investment on the part of MCI.

This option requires the replacement/upgrade of existing workstations with new equipment (increased memory), and an entire O/S migration from existing software. It would standardize the O/S for the database server, LAN servers, and all workstations to Windows NT. The client microcomputers for SSD (and any other users) must be 32 bit systems, Pentium processors, with a minimum of 16 MB memory and 1 GB storage. They must be configured fo network access and provide access to electronic mail, business suite applications, a relational database, message system software, World Wide Web access, and customer service/help desk applications. This option is depicted in Figure 4.

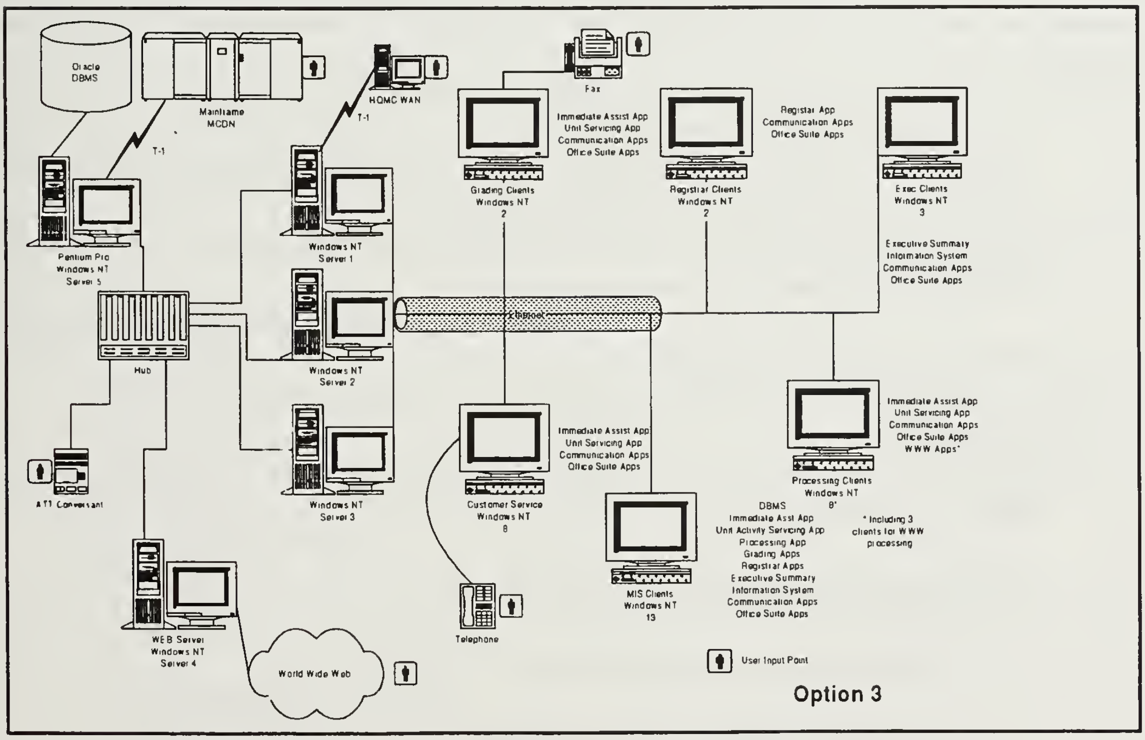


Figure 4. Option Three Technology Architecture.

b. Migration Issues

This option represents the most complex change and can therefore be the most problematic. It will require MIS personnel to learn a new O/S, new hardware, new applications, and a new DBMS simultaneously. It will be the most difficult migration implementation. It will require a sophisticated gateway for transition from MPE/iX Turbo Image to a Windows NT based version of Oracle® 7.3.

This option does embrace emerging technology and will place MCI in the front of the mainstream of technological implementation. There is significant likelihood that the USMC is transitioning to NT as a server O/S, and this would position MCI to have an O/S that was standard compliant and enterprise wide (mail server, web server, file server, database server all on the same O/S). This would simplify middleware issues for the system.

c. Advantages of Option Three:

- Oracle® corporate interest in NT developments
- Position for standardized operating systems
- Current direction of industry
- Newest technology
- Hardware costs are comparatively low
- The most responsive option to future changes

d. Disadvantages of Option Three:

- Very steep learning curve for MIS personnel
- Significantly more complicated migration

- Client/server middleware reliability issues
- Unproven scalability
- Many more variables--highest risk

D. RELATION OF TECHNOLOGY PLATFORM TO APPLICATIONS AND BUSINESS FUNCTIONS

The business locations and their definitions are less germane to this particular application of the EAP method than they would have been for a true enterprise level redesign. Since the MCI project is only being implemented at a business unit level, the location of all pertinent business functions is the Student Services Department of MCI. The Deputy Director, the MIS Dept., the Courseware Development Dept., and all the various conceptual sections of SSD will need access to the information and its applications. While the data will physically reside on a central server, different applications will reside on different clients. Functionally, the SSD is broken down into billets which perform various duties, and for the purpose of this discussion will be considered "sections" of SSD, although there may be no realistic corresponding organizational structure for these conceptual business functions.

All sections of the SSD are located in the confines of the SSD operational area at MCI headquarters. The conceptual business functions performed under the SSD are:

- Grading materials
- Process Unit Activity Report
- Enroll/Disenroll Students
- Perform Registrar functions

- Process Incoming mail/e-mail
- Process Incoming phone calls
- Provide input to error listing for MIS

While these business functions are all the responsibility of SSD, they may be subdivided into notional sections that are staffed by different personnel but retain an assigned business tasking. The registrar section performs registrar functions. The Immediate action section provides the processing of incoming phone calls. The grading section performs the grading of materials. The process section provides for the enrollment/disenrollement of students and processes incoming mail/e-mail. The Unit Servicing section processes UAR's and provides input to MIS for the purpose of the error listing report.

Since these sections are somewhat notional, one person may actually perform the duties of numerous sections, but it is conceptually useful to divide them functionally for the purpose of providing technology architecture to support these capability requirements. The assignment of proposed applications to physical locations within SSD in order to support the business processes developed in [Ref. 4] is a natural extension of the process methodology used in this report. The physical distribution of clients is depicted in Figure 5. As indicated in the final report to MCI [Ref. 4], the clustering of the CRUD matrix developed for processes and their related entities reveals the applications required by each sub-unit of SSD. These applications must be mapped to client workstations in the MCIAIS II network. Figure 5 shows these physical locations and their distributed application requirements, while Table 1 presents these results in tabular format.

and their distributed application requirements, while Table 1 presents these results in tabular format.

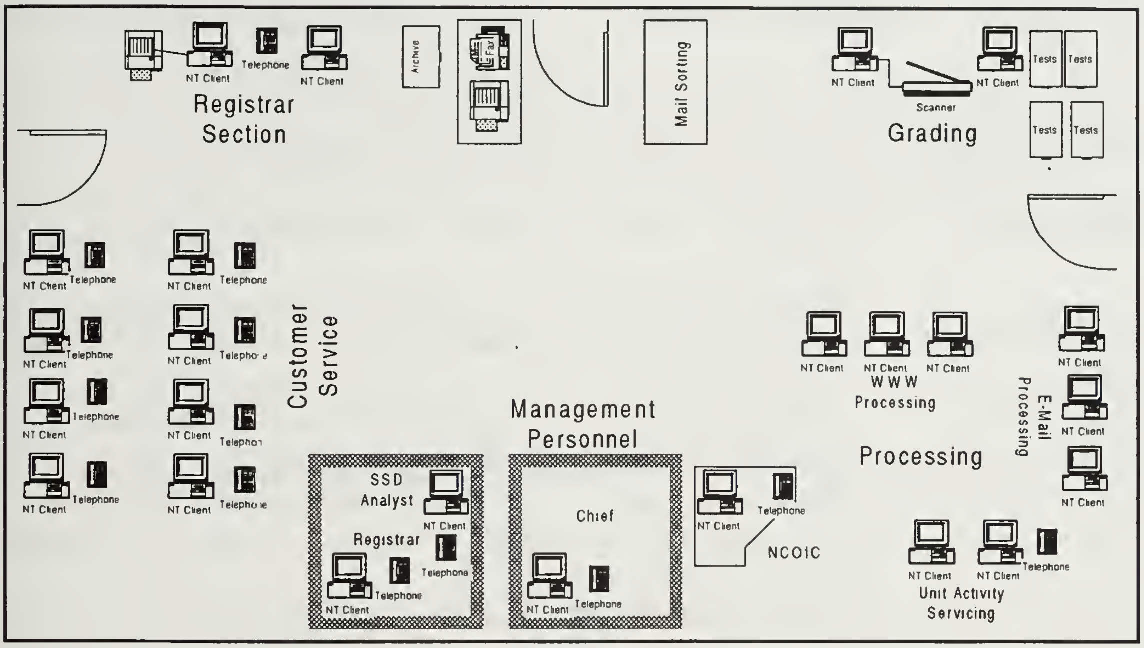


Figure 5. Physical Distribution of Clients by Business Area.

The intent of Figure 5 and Table 1 is to illustrate the logical distribution of applications to the physical assets recommended for SSD and to determine their approximate location. The exact location of the client workstations or the resulting work flow issues raised as they apply to the "To-Be" system are left to the system implementers.

One technology consideration deliberately left out of this architecture definition per the instruction of MCI was the potential for World Wide Web enrollment/assistance requirements for the Processing subsection. Although this requirement has been intentionally excluded due to MCI planning limitations, it should be addressed by development planners for MCIAIS II as soon as goals and plans for use of the World Wide Web are developed.

Location	Entity/Data	Applications
Deputy Director	Executive summary	Executive Summary Information,Office Suite,E-mail
MIS	All	DBMS, All Customer Apps
SSD Mgmt	All	Exec Summary Info, Grading, Student Servicing, Unit Servicing, Office Suite,E-mail
Registrar Section	Student, Course, Program	Registrar, Office Suite, E-mail
Grading Section	Student, Course, Exam, Exam_Answer_Key	Grading, Office Suite, E-mail
Immediate Action Section	Student, Course, Customer, Call,	Student Servicing, Office Suite, E-mail
Unit Servicing Section	Student, Course, Program	Unit Servicing, Student Servicing, Office Suite, E-mail
Process Section	Student, Course, Program, Exam, Exam_Answer_Key	Student Servicing, Office Suite, E-mail

Table 1. Location/Entity/Application Listing.

V. THE MIGRATION PLAN

The formation of the technology architecture is a critical step in reaching the desired goal of a new information system, but a migration strategy to reach that goal is just as important. This chapter will discuss migration strategies in general, some of the considerations for choosing and defining a migration strategy, and finally describe the migration strategy for MCI.

A. MIGRATION STRATEGIES

1. Introduction

The MCIAIS redesign was undertaken because of the inherent inability of the existing legacy system to support current and future business practices. As is typical of legacy systems, the underlying data structure is poorly documented, applications have been patched repeatedly, maintenance costs are high and documentation of modifications to data structure has not been consistent. Further, the current IS budget is being spent on the maintenance and operation of the current legacy IS and there is not enough funding to finance the required upgrade to migrate the system. Brodie and Stonebraker [Ref. 5] coined the term *IS Apoplexy* to describe this situation. To complicate matters further, the lack of flexibility resident in a legacy system prevents the IS department from integrating new technologies with the legacy IS.

Migration to open systems is a topic of much discussion in technology publications. By recognizing the need to migrate from the current legacy system, MCI has taken a strong positive step towards ensuring their future viability as a service provider to their customers, the United States Marines.

Migration of legacy systems to contemporary, open architectures is a difficult task with many unanticipated levels of complexity. The methodology used to support migration is based on two fundamental approaches to the issue of migration. The first approach, known as the cold turkey approach, is instantaneous transition from the legacy to contemporary system and the second, known as the chicken little approach, takes a more cautious and incremental approach to migration. These two approaches are discussed in the following sections.

2. Cold Turkey Approach

The Cold Turkey strategy is so named because it is analogous to a smoker who quits "cold turkey". It means that the target IS will be developed from scratch using modern software techniques and new, different hardware. There is no interaction between the legacy system and the target IS, and the intent is that when the target information system is ready, operations will be cutover from one system to the other. This cutover is inherently risky, and the overall strategy of cold turkey has other impediments to success as identified by Brodie and Stonebraker [Ref. 5]. These impediments include:

- The system must be better, not just newer.
- Business conditions don't stand still while the target system is being developed.
- There are rarely specifications documented on the legacy system. Often the only documentation is the code itself.
- There are numerous undocumented dependencies between applications.
- The size of legacy system data may prevent timely cutover on mission critical systems.
- Managing large projects is hard.

- Lateness is seldom tolerated.
- Large projects tend to bloat.
- Homeostasis is prevalent in the IT world. Fear of change, new techniques and new technologies contribute to enormous resistance to cold turkey migration.
- Analysis paralysis sets in. Migration doesn't begin until you understand the legacy system.

The primary shortfall of cold turkey migration is that there is no period of gradual adjustment or flexible adaptation to the new system. The actual cutover is incredibly risky -- the mission critical information system will be turned off and operations will begin on a new system with the potential for untested bugs. Very few organizations can afford this level of risk. While it is not unreasonable under the current operating environment for MCI to suspend operations for a short period while cutover and testing occur, the many other disadvantages make this option unattractive, and therefore is not recommended for MCI. An incremental strategy is more suitable as discussed in the following section.

3. Chicken Little Approach

The incremental migration strategy is named for the fabled character with the cautious, conservative view of the world. Brodie and Stonebraker feel that this type of cautious and conservative approach is necessary to ensure the smooth running operation required of a successful migration effort. According to this strategy the legacy system is migrated in place in small incremental steps until the desired long term objective is reached. Each step requires smaller resource allocation and less time to complete than a cold turkey approach. Because the strategy is incremental, the risk associated with the migration is also incremental and is divided between each of the steps. The smaller the increment of migration, the smaller the risk to the system, but also the smaller the potential gain for the system owners. By dividing the risk, the organization is insulated from the effects of the failed migration in total. If a chicken little step fails, only that step has to be repeated. Brodie and Stonebraker's eleven chicken little steps (discussed later in detail) are as follow:

1. Incrementally analyze the legacy IS.
2. Incrementally decompose the legacy IS structure.
3. Incrementally design the target interfaces.
4. Incrementally design the target applications.
5. Incrementally design the target database.
6. Incrementally install the target environment.
7. Incrementally create and install the necessary gateways.
8. Incrementally migrate the legacy database.
9. Incrementally migrate the legacy applications.
10. Incrementally migrate the legacy interfaces.
11. Incrementally cut over to the target IS.

Like any other complex plan or strategy, migration requires an underlying set of fundamental principles to guide the effort towards success. Some examples provided by Brodie and Stonebraker [Ref. 5] of migration requirements for a typical legacy system include:

- Migrate in place.
- Ensure continuous, safe, reliable, robust, ready access to mission critical functions and information at performance levels adequate to support the business workload.
- Make as many fixes, improvements, and enhancements as reasonable to address current and anticipated requirements.
- Make as few changes as possible to reduce migration complexity and risk.
- Alter the legacy code as little as possible to minimize risk.
- Alter the legacy code as required to facilitate migration.
- Establish as much flexibility as possible to facilitate future evolution.
- Minimize the potential negative impacts of change, including those on users, applications, databases, and in particular, on the ongoing operations of the mission critical IS.
- Maximize the benefits of modern technology and methods.

By addressing these requirements properly, a framework for successful migration can be developed for MCI. While this list may not be inclusive of all migration requirements, it reflects a good framework for any migration effort.

4. Complexity of Architectures

As important as the migration strategy and incremental philosophy may be, the inherent complexity of the legacy IS architecture must be considered when conducting migration planning. As stated by Brodie and Stonebraker, any IS can be decomposed into three basic levels. These are the interface components, the application components, and the database components. The degree of dependency between these three components will impact significantly on the ease of migration. Legacy systems that were designed without the benefit of modern analysis, design, and programming methods lack the modularity and independence of modules that are incorporated into current structured development techniques.

Brodie and Stonebraker categorize systems into three ascending levels of complexity: decomposable, semi-decomposable, and nondecomposable. The type of system structure determines the migration steps and actions taken under those steps.

A decomposable system is the easiest to migrate because the interfaces, applications and database are separate with well defined interfaces between them. Interdepencies do not exist between the applications and the applications interface only with the database. There is no hierarchical structure among the applications, and all modules are documented with their relationships. Unfortunately, this structure rarely exists in legacy systems.

The next type of system in ascending order of difficulty for migration is a semi-decomposable system. In this type of system, the user interfaces and system interfaces are separate modules, but the applications and database cannot be separated. This dependency may be the result of older development methods, poor systems engineering, or nonexistent documentation.

In a nondecomposable system, the entire system is built without documented dependencies. It may be viewed from the migration planner as a "black box" with no separate modules. A system that combines characteristics of both decomposable and nondecomposable systems is labeled a "hybrid" system. This type of system indicates that some applications have no modular development, and contain undocumented dependencies with the database. Some other applications are decomposable at some level, modular and not dependent on the database or other applications.

Based on its characteristics, MCIAIS can be classified as a semi-decomposable system. On one hand there is a high level of undocumented dependencies between various applications and the database level that create a large obstacle in decomposing the system for easy migration. On the other hand, many system interfaces and user interfaces have been developed recently and can be considered separate modules.

B. MIGRATION CONSIDERATIONS

1. Role of Gateways

A key requirement of successful migration is to view and operate the legacy system and the target system as a composite information system, providing the critical business functions together until migration can be completed. The key component in the ability to operate in a collective environment is a gateway. A gateway is specially developed software that provides an interface between the two systems while insulating certain components from changes being made to other components. The gateway must be able to translate data and function requests between the various components, and coordinate updates so that data remains accurate and consistent in both the old and target IS.

The gateway will need to insulate the legacy interface from changes made to the other parts of the legacy IS. This insulation should provide transparency to the user, who will not be able to determine whether the legacy IS or the target IS is executing certain functions. This is a critical feature of the gateway that must be available in order to successfully implement the incremental approach to migration. Queries and updates must be translated between the old and new systems, maintaining accuracy and consistency of data. The transaction management capability of the gateway must be able to guarantee to the user that the integrity of the mission critical data is being maintained during the migration process.

Gateways can be functionally classified as either forward, reverse, or bi-directional, depending on their direction of translation. Forward gateways enable the legacy interfaces to access data on the target system. Forward gateways are used for translating instructions from the old system *forward* to the target system. Reverse gateways direct data in the opposite way. Instructions received on the target interface are translated in *reverse* to the legacy system. Bi-directional gateways support translation in both directions, and are intuitively the most complex. During the migration process, functionality of both forward and reverse gateways is required at some point to incrementally migrate systems. With a bi-directional gateway, these functionalities are incorporated in one product.

The bi-directional gateway supports the concept of parallel operations, i.e., the two databases operate simultaneously, replicating transactions received on either database over to the other database. To successfully implement incremental migration, MCI must use a gateway that will support the parallel operation concept. There will be a requirement to synchronize the data between the two database systems, and a requirement to replicate all transactions from either system interface during the migration implementation.

Operating two heterogeneous databases simultaneously is not trivial. For the user to remain unaffected by the migration, there must be complete transparency between the two databases. Current gateway products provide for simplified procedures and programming which have significantly improved the capability of interaction between databases. A parallel operation must be able to keep the data between the heterogeneous databases synchronized, but in doing so, allow the gateway to be the central point of management for updates to data structure and location.

Expectations of the gateway product should include a capability or some type of dynamic dictionary mapping to ensure the data structure of the legacy and target systems are synchronized, and must be able to guarantee transactional integrity so that communication losses do not result in inconsistent data between the heterogeneous databases. Automatic replication is another expectation of gateway products. Any gateway used should be able to automatically replicate data from a specified source (such as tables populated from MCTFS) and ensure that the heterogeneous databases are all synchronized with the input data from the replication source.

The placement of the gateway in the architecture affects the complexity of the migration effort and is determined by the potential decomposition of the legacy IS (see Figure 6). If the system is decomposable, the gateway can be placed between the applications and the database, and functions as a database gateway. For a semi-decomposable system, the gateway is labeled an application gateway, and must be placed between the interfaces and the rest of the legacy IS. The last type of Gateway is called an IS gateway, because it interfaces the entire legacy IS with the target IS. Because of the inherent complexity of nondecomposable system migration, this gateway type is the most complex.

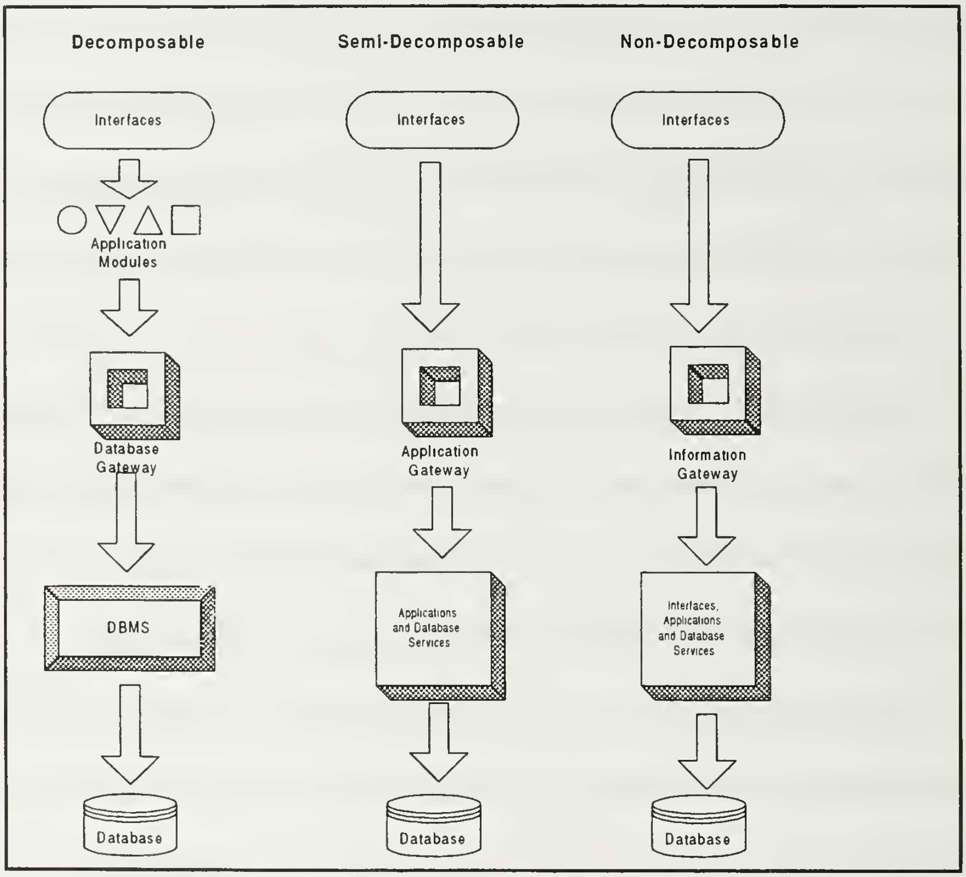


Figure 6 . Placement of Gateways.

2. Migration Cutover

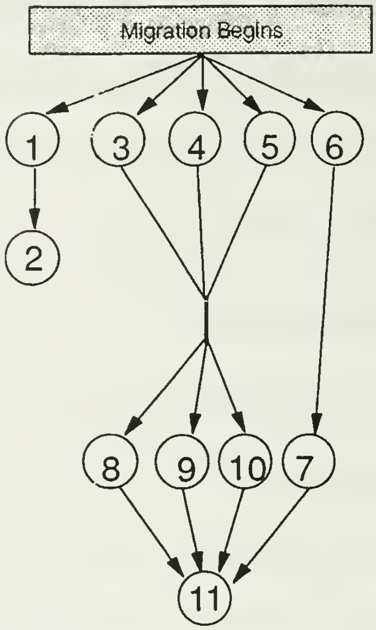
Cutover is the key difference between cold turkey and incremental migration. Because the cutover of the system carries the highest level of risk exposure for the organization, it must be considered incrementally in concert with the rest of the migration philosophy. The decision must be made to incrementally cutover to the new IS as the target IS applications are tested and approved. These cutovers can be accomplished by organizational applications or by logical units of functionality. Decisions for incremental cutover can be made for the applications discussed in Chapter IV for the physical locations, or can be applied to logical applications across their physical boundaries.

3. Version Control and Configuration Management

The entire migration process must include consideration for the requirements for improving version control and configuration management of both the legacy IS and the target IS during all steps of migration, but especially during the cutover step. The developers must consider version control of code that is being developed to run on two separate platforms, and maintain current documentation on this temporary code to ensure that failed steps can be properly analyzed with complete documentation of the transition code.

C. MIGRATION STRATEGY FOR MCI

As discussed before, the chicken little strategy is the recommended strategy for the incremental migration from MCIAIS I to MCIAIS II. Figure 7 (reproduced from *Migrating Legacy Systems*, pp 32, [Ref. 5]), displays the potential paths for the steps used during migration. Many of these steps can be executed in parallel, and the simultaneous completion of the different steps can reduce the overall migration time frame. Each of these steps in the incremental migration strategy is discussed in detail in the following section.



1. Incrementally analyze the legacy IS.
2. Incrementally decompose the legacy IS structure.
3. Incrementally design the target interfaces.
4. Incrementally design the target applications.
5. Incrementally design the target database.
6. Incrementally install the target environment.
7. Incrementally create and install the necessary gateways.
8. Incrementally migrate the legacy database.
9. Incrementally migrate the legacy applications.
10. Incrementally migrate the legacy interfaces.
11. Incrementally cutover to the target IS.

Figure 7. The Steps of the Chicken Little Migration Plan.

1. The 11 Steps

As indicated earlier, there are eleven steps that establish the framework for an incremental migration plan. What follows is a brief discussion of those steps, and their application at MCI.

a. Incrementally Analyze the Legacy IS

This first step is especially important to identify as many dependencies between the applications and the database as possible. A strong "As Is" model will greatly increase the probability of success for the new IS. It is important, however, to resist analysis paralysis that is usually the result of an overwhelming feeling that one needs to conduct further analysis to completely understand the legacy system. This effort is considered complete once the essence of the business rules and data requirements have been captured. The majority of this step has been completed for the SSD portion of

MCIAIS II. The design team must still conduct some analysis to verify and validate the delivered models and documents.

b. Incrementally Decompose the Legacy IS Structure

The amount of effort here must be governed by the anticipated time expected to conduct the migration. If the legacy system is intended for relatively short use (as at MCI) then it probably does not make sense to put a great deal of effort into rewriting code on the legacy system to facilitate decomposition. The migration team needs to identify key dependencies that will impact on the gateway operations, and recode them to support the gateway requirements for parallel operations. In the case of MCIAIS I, the in-house programmers should be familiar enough with the legacy code to assist a migration team in the necessary redesign of the code to support this decomposition.

c. Incrementally Design the Target Interfaces

For this step, the planner must decide which, if any, of the user and system interfaces on the legacy system will be migrated to the new system. For MCIAIS I, the user interfaces will be developed from scratch using functional requirements developed from user input and the information required as demonstrated in legacy user interfaces. The interface is poor enough on MCIAIS I that significant user resistance is not anticipated to the change of user interface. The graphical user interface being developed will be much easier to use, and the gateway will allow the users to access old data from the legacy system while the new system is under development.

d. Incrementally Design the Target Applications

The target application designs are completed to conform with the business rules adopted and defined under step one. Any business rules adopted for applications must be enforceable on the target environment. New business rules that did not exist in the old system bring with them some element of risk to add them to the new application. On the other hand, the ability to institute these rules may be a predominate factor in migrating the system in the first place, so the evolution of business rules in the new applications must be carefully considered. Adding new data entities, for instance, may be much higher risk activity than simply installing a graphical user interface on the old system, but reflects a fundamental requirement that is driving the migration process, as is found in this project. For MCIAIS II, the target system would not be acceptable without these higher risk modifications. The Oracle® DBMS provides a high level of functionality to support the business rule requirements in the form of triggers and stored procedures.

e. Incrementally Design the Target Database

The database model should be built with a thorough understanding of the data requirements of the organization. These requirements most likely exceed the data currently stored in the legacy system. The design team must be prepared to provide extensive documentation and version control to ensure the documentation lapses of the legacy system are not duplicated. This is especially important as application development modifications occur later in the implementation phase. Tools for database development should be incorporated to insure synchronization between the database models and the actual program. As discussed in the final project report [Ref. 4] the ER*win*® product provides a high level of synchronization capability between the data model and the Oracle® operational schema.

f. Incrementally Install the Target Environment

The target environment must be defined based on the aggregate requirements of the total target system. Development should begin as if the target environment is being developed from scratch without the benefit of any code or information from the legacy system. One of the most visible changes in the target environment will be replacing the "dumb terminals" with either networked client PC's or workstations. Once the target environment is selected, it should be installed incrementally to minimize risk and give users time to adapt to the new system. This portion of the migration is very expensive, as it involves replacing all of the hardware and system software for each user. It is also the most visible element of the migration to the system users, as it takes place on their desktop. Because this step involves the adaptation of users and management, it can easily become the longest step in the migration process, hampered by human politics and psychology. The majority of the hardware costs for this step have already been incurred for the users. Minimal costs for additional upgrades as required should be expected and included in future budgets.

g. Incrementally Create and Install the Necessary Gateways

If commercial products are not available for this step, it will be the most technically challenging portion of the migration. If the organization can obtain commercially produced gateways in support of their migration plan, chances for success will be greater than if the gateways must be developed from scratch. Increasingly, more commercial products are available, as are experienced migration specialists to support the migration process. In conferring with one of the methodology authors, he indicated that gateway development has improved and many more successful migration efforts have been recorded [Ref. 6]. Statistics on these efforts were unavailable at the time. The availability of commercial gateways and potential solutions for this effort will be addressed later in this chapter, but their ability to satisfy this requirement is fully anticipated.

h. Incrementally Migrate the Legacy Database

With a gateway in place, one-time migration of the data can begin. Depending on the strategy chosen, the organization can choose to populate the target database once, and discontinue active use of the legacy database, or the organization can take advantage of the replication features of the gateway and use whichever portion of the database best supports the logical migration of user applications.

The implementation of a one time migration is more closely associated with a cold turkey migration strategy, but can be adapted to an incremental migration strategy as well. For MCI, a data migration plan using a gateway is the preferable method.

Data migration follows three basic steps: 1) Identification of attributes currently available; 2) Mapping and migration of currently available attributes to the newly developed database; and 3) automatic input of attributes not currently available at some future time (once they become available through automation and redesign of other MCI systems and departments), or manual input of attributes not currently available and not likely to be made available through the automation of other MCI systems or departments.

i. Incrementally Migrate the Legacy Applications

For steps 9, 10, and 11 in a semi-decomposable system, the development team can choose to conduct all three steps together, or attempt to find logical divisions in the applications. By definition, the semi-decomposable legacy system resists decomposition of these layers, so the migration will necessarily be dependent on the ability of the development team to separate the applications from the data. Since no actual application code is being migrated for this effort, the application migration is addressed in natural divisions of work/organization. As the development team completes and tests applications, they can submit them to the SSD sub-units that are the intended users of the applications.

After iterative user testing, these applications can be migrated to the user in segmented sections, such as migrating the processing application first, then the grading application, etc., or the total application for a functional area can be migrated to some of the desktops. For example, some of the Customer Service desktops could be migrated, leaving dumb terminals on other desktops to facilitate more thorough testing and user adaptation. This plan will have to be developed by the design and implementation team after some of the application development has been completed.

j. Incrementally Migrate the Legacy Interfaces

The dumb terminals represent the primary user interface with the system, and can be replaced on some of the desktops immediately upon new interface development. Minimal numbers of dumb terminals may need to be retained for periods of non-availability during development of the target IS. The gateway will replace the primary application interfaces and maintain the required synchronization of the legacy and target databases.

k. Incrementally Cutover to the Target IS

As discussed previously, the cutover can be accomplished by organization section applications or by logical units of functionality. The key to this step is taking the cutover in small enough pieces to manage effectively. It is important that no major portions of the system be cutover simultaneously. Doing so will raise all of the risk factors associated with cold turkey migration. The cutover can be integrated with the test plan to accomplish the module cutover in conjunction with successful testing. By limiting the size of cutover modules, it will also give users more time to train for and adapt to the new system.

2. Hardware Migration Issues

The next area for development of migration strategy is the consideration of hardware related issues. All three options, discussed previously in Chapter IV, require hardware changes that must be incorporated into the overall migration strategy. MCI currently is operating a Hewlett-Packard 3000 model 957. This is the baseline hardware platform. The migration path for hardware is determined by which of the three previously defined options is chosen.

a. Option One

This option requires either upgrading the current minicomputer by adding memory and storage capacity, or migrating to a HP 3000 model 200, which we believe is an overkill for the requirements of this migration effort. This minor upgrade option provides a familiar hardware environment, and would have the lowest impact and lowest risk of any of the options.

b. Option Two

The second option provides for transitioning the current system to a HP 900 model 969 minicomputer. This is a relatively familiar hardware environment, but does have significant hardware distinctions from the HP 3000 environment. This computer model represents a significant upgrade in processing capacity and will provide computing power for MCI for many years. The overall cost of this option is very high, which is categorized as medium risk.

If this option is chosen, migration will require moving the existing legacy system from the old hardware to the new hardware. Although this step represents an operating system change, in addition to a hardware change, technical representatives from the vendor are expected to support the effort of porting the legacy system in its current form to the new hardware. The new hardware base will need to be integrated with MCI's local area network and will also need connectivity to the MCTFS database.

c. Option Three

The third option represents the desired end state for MCI and is based on an Intel microcomputer running on a Windows NT server. This configuration represents a new hardware suite which will require additional training and vendor assistance during the migration effort. MCI must be willing to support comprehensive training of MIS personnel in both hardware and software to ensure the success of this option. Because of the steep learning curve for MIS personnel, this option is best suited for incremental migration over a period of time and preceded by option one. Selecting this option initially is categorized as very high risk.

3. Software Migration Issues

All three options address a change in the DBMS software, and two of the options discuss changes in operating system software. These changes must be incorporated into the overall migration strategy.

a. Option One

Option one does not require operating system migration. Retaining the current operating system greatly reduces migration risks by providing personnel with a familiar system environment. Current applications can be retained, but the current TurboImage database will require migration to an Oracle® relational DBMS, and this step is common to each option. The straightforward advantage of the easy software migration path is enough to make option one the best choice for MCI, as well as the lowest level of risk.

b. Option Two

Option two provides for a transition of operating systems from MPE/iX to HP/UX. Because MCI personnel are not trained in HP/UX there will have to be significant investment in personnel training to facilitate the operational knowledge required of system administrators of a UNIX based system. Current applications will have to be reprogrammed for HP/UX or rewritten at an unknown cost. While UNIX platforms are certainly open, transition to UNIX minicomputers is not the direction of the computer industry. Because of the training costs and application software costs, this operating system transition is characterized as high risk, and may be a potential point of failure for this project.

c. *Option Three*

Option three is a transition to a Windows NT operating system based on a microcomputer platform. While it carries the same risks as option two, the low cost of the hardware platform and operating system allows for small scale implementation, thus providing more time for MIS personnel to be trained and prepared to run this system. Because of the overlap in operating systems between all the servers at MCI, it would be possible to train all MIS personnel in the operation of all the servers at MCI, instead of the current division between the minicomputer personnel and microcomputer personnel. A standard operating system represents consolidated training costs, and lower maintenance costs. Integrated properly with Option one, this path is the right long-term solution for MCI.

4. Other Software Issues

If MCI elects to transition to a UNIX environment (HP/UX), it will entail migrating the current database and all current legacy applications from MPE/iX based software to HP/UX based software. MCI will need to obtain vendor support for migration of any applications that are not supported by Hewlett-Packard. In order to ensure operational stability of the new version of the legacy system, this step needs to be completed before development of the target system begins on the new platform. This will create a critical time delay for MCI during a period when personnel are attempting to learn to manage a new operating system and integrate new hardware systems.

Failure to ensure system stability could seriously jeopardize the entire target development effort. After the operating system and new hardware have been successfully installed, the next step is to migrate the HP/UX TurboImage database to a relational database. This step will enable MCI to incrementally migrate from a flat file system to a relational table based database. MCI has two basic choices for incremental migration: either migrate the HP/UX TurboImage database to HP/UX Image/SQL database, then migrate the Image/SQL database to an Oracle® database or migrate from TurboImage directly to an Oracle® database.

The first option is consistent with an incremental migration, and will allow MCI to take advantage of HP support for both products, while migrating the core data into a relational format based on the data model developed by the NPS team. There are commercial firms which specialize on this type of migration, and can be consulted to aid MCI in this effort. Upon successful completion of migration to Image/SQL, MCI can continue to plan for eventual migration to an Oracle® relational database management system. Oracle® Corporation supports a commercial gateway for migration to Oracle® Server 7.3 from Image/SQL databases. This gateway will be discussed in some detail later in this thesis.

The second option is also consistent with incremental migration and eliminates the costly requirement to purchase the interim software for Image/SQL. By purchasing a direct gateway between TurboImage and Oracle® databases, the migration transition timeline can be reduced, redesign of the system can be simplified, and support for longer term migration will be enhanced. Commercial solutions are available for this type of migration and are discussed later in this chapter.

D. CONCLUSIONS

Upon consideration of the issues relating to migration, several conclusions were reached that directly impact this effort. These issues need consideration by MCI in order to maximize the benefit of system migration.

1. Preventing New Legacy Information Systems

Unfortunately, the new system of today will become the legacy system of tomorrow. To ease the pain of future migration, modern analysis, design, and implementation techniques should be employed. By designing clearly documented modular applications, MCI can prevent the reoccurrence of the very difficult migration path that exists today. As indicated in Chapter IV, the target IS must be designed to meet the architectural principles under which this project was begun. MCIAIS II must be portable and able to support future rightsizing efforts. No one can predict with certainty the future business requirements for MCI, nor the future capabilities of technology, nor how the former will interface with the latter. By developing an open, scaleable, portable system that leverages flexibility to a maximum, MCI can position itself to favorably deal with the future.

At this time there is no way to predict the role of the World Wide Web (WWW) on how MCI will eventually support students. Rapid development in distance learning techniques and methods may render MCI's current role obsolete in a few short years.

MCIAIS II must be flexible enough to adapt to the direction of technology and changing roles and missions.

2. Continuing Migration to PC Platforms

In order to maximize the benefits of flexible technology and capitalize on industry direction, MCIAIS II can only be seen as an interim step in a continual migration process. Current trends in the development of microcomputer servers indicate the development of four, six and eventually ten processor servers which will exceed the planned HP 9000 minicomputer's processing capabilities for much less capital investment. The ability to distribute the servers and the application processing provides significant increases in flexibility and availability, while providing decreased risk. The eventual migration to microcomputer based systems will align MCI with industry direction, and minimize capital expenditures for the future. It would be in the best interest of MCI to begin this migration planning immediately, potentially purchasing the first server in 1998 to begin development and training on the microcomputer platforms. Individual applications can be migrated seamlessly to other platforms using the Oracle® Server software.

The biggest danger of the purchase of the new HP 9000 is that MCI planners will tie future development of the MCIAIS to the sunk cost of this platform. In choosing this platform, MCI did not address the full life cycle cost of this system. The high maintenance costs of this system suggest that MCI should migrate off of this platform at the first available opportunity. To facilitate this migration, the earliest implementation of a microcomputer based server for database distribution is strongly recommended.

3. Potential Commercial Solutions

There are two potential commercial solutions which merit further discussion as a result of this research. The first potential solution is an Oracle® Gateway Solution, the second is a third party vendor solution.

a. Oracle® Gateway Solution

This gateway could only be used if a transition from TurboImage to Image/SQL is conducted. This product is the Oracle® Transparent Gateway for Image/SQL. All indications point to a capable gateway product, but one that only works in the SQL/Table based environment. Oracle® Corporation markets and sells customized gateway solutions that claim to satisfy the same requirements as the commercially available gateways, but at a greater cost due to the customized nature of the product. For the discussion of this gateway, the information centers on the capabilities available in the transparent gateway available for Image/SQL to Oracle® DBMS. This product will support read and write access to the Image/SQL database from the Oracle® applications. Image data can be migrated periodically or in a single population move to the Oracle® database. This gateway solution will support the migration strategy endorsed in this chapter, but will require the installation of the Oracle® server product as well as previous migration of the database to Image/SQL before the gateway could be used. Oracle® Corporation indicates that it is investigating potential solutions to migrate directly from TurboImage, but none are commercially available at this time.

The Oracle® gateway solution claims to support transparency, dynamic dictionary mapping, and complete synchronized automatic replication. Oracle® gateways implement transactional integrity through a "two phase commit" process. This mechanism insures consistent data by not allowing writes or updates to the Oracle® database to be "committed" or completed without verification that they have posted to the legacy database as well (see Figure 8). This integrity and replication capability would be especially valuable in mirroring the data found in the MCTFS tables, the conversant tables, the legacy tables and the target tables. This gateway capability is especially interesting for this migration project, but the costs and risks of additional migration to Image/SQL outweigh the potential gains. A custom gateway solution from Oracle® that provides the same functionality as the Transparent Gateway for Image/SQL should be thoroughly investigated.

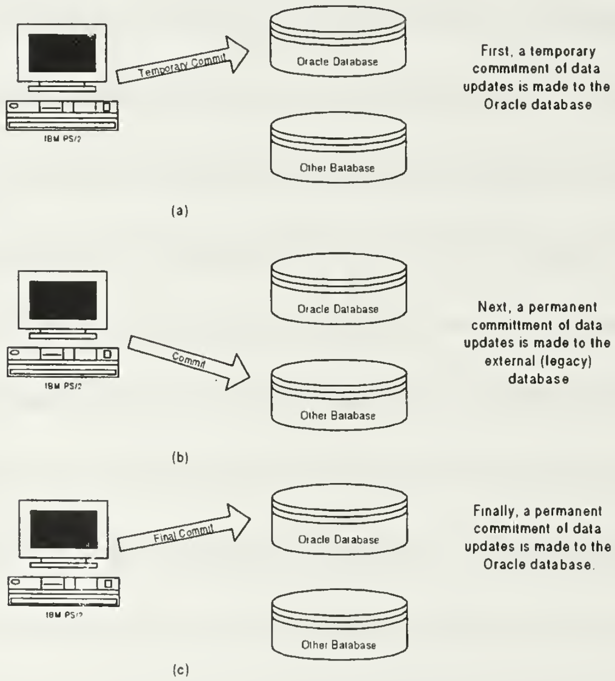


Figure 8. The Oracle Two Phase Commit Process.

b. Third Party Gateway Solution

During the evolution of this study, only one potential commercial solution has been found for the migration of data directly from TurboImage to Oracle®, this addressing the current needs of MCI. This potential solution, available from Starvision Inc., is a gateway application that provides an out of box gateway solution for TurboImage to Oracle® translation. There are two variants: an unidirectional gateway and a bi-directional gateway. The unidirectional gateway allows users of the TurboImage applications to update the Oracle® database or allows users of the Oracle® applications to update the TurboImage database, but not both. To get the bi-directional functionality, the bi-directional gateway must be purchased. This product provides full replication of data and best supports the parallel migration plan in its current form. The Starvision product was developed internally for HP and has been improved and marketed separately. It specializes in MPE/iX TurboImage gateway capability, and seems to be the most logical fit for the existing requirements at MCI. It is adaptable to other environments and could be used to support HP/UX TurboImage as well.

Additional advantages of this product on an HP 3000 include significantly enhanced performance and throughput of data due to "Standby Agents" which are stand by processes that are run before the actual requests are made by the user. This results in predefined data views being available much more quickly. The Starvision gateway is compatible with most development tools chosen for client side development, and is fully integratable into an intranet environment due to its open system design. This solution, or any other gateway that will support this type of requirement is the type of middleware needed to adopt an incremental migration from the current database to an Oracle® database solution.

VI. TRANSITIONAL ANALYSIS OF THE MCI MIGRATION EFFORT

One of the most critical, yet often overlooked areas of potential failure in a complex migration effort is the proper management of the transition itself. Failure to take into account the human factors of migration is potentially the most problematic area that planners will have to overcome. This chapter will address some of the concerns people have when encountering change, how these concerns have applied to the MCI project, and some actions that planners and leaders can take to ensure the smoothest possible transition to a target system.

A. OVERVIEW OF CHANGE

The business world of today is rife with examples of organizations that have failed to meet the challenge of change. A successful organization must initiate change, not just react to it. This means the organization must know its people, their capabilities, their level of training, and their basic disposition towards change. All too often, change for the organization is a reaction to crisis, and must be implemented in the crisis state. By strategically incorporating planned change into long term systems migration efforts, the organization can prepare its people for the change ahead, and allow them to shape the methods and means of transition.

1. Definitions and Descriptions

The Harvard Business School has defined change as "a planned or unplanned response to pressures and forces [Ref. 7]." These "pressures and forces" have always been around, but one could argue that the dawn of the information age has increased their intensity and frequency. Organizations operating and competing in the information age must be prepared to keep ahead of the forces shaping the information world. To prepare an organization for this newly required mind set, leaders must incorporate an active transition management strategy to cope with the inevitable resistance to change.

In his book on the payoff of technology for today's CEO, Hoffman has a pointed address on the resistance to technological change.

> New uses of information technology require people to change the ways in which they do their jobs, and most people resist change. They resist change because they are comfortable with their current ways of doing things, because they know that the change process itself will be painful, and because they are unsure about how the new situation will work out. This is particularly true of changes that add information technology to jobs that did not previously use it. Many people are uncomfortable with computers, some to the point of fear. People who perceive themselves to be successful in their jobs are particularly prone to resist changing the ways of working that helped them achieve their success [Ref. 8].

The last sentence of Hoffman's statement is particularly important in today's world. Increasingly, successful people identify themselves with their careers. Some studies demonstrate that this behavior is spreading from the realms of the elite to the lower levels of organizations as well. Nicholson states, "The more challenging, complex, and demanding are our occupations, the more likely we are to think of our careers not as part of our lives, but *as* our lives [Ref. 9]." If this is true, then surely the stakes have been raised in the transition state. We are not just changing our jobs, we are changing our lives.

2. The Need to Hold On

People resist change for many reasons. It might be due to lack of information about the change, or perhaps it originates from fear that the change will bring with it a loss. One of the first aspects of resistance to address is the need to hold on. This means holding on to the comfort zone, the attitudes and behaviors that have shaped the successful work experience, even the machinery used to accomplish this work. As long ago as 1961, James Baldwin wrote in his book:

> Any real change implies the break up of the world as one has always known it, the loss of all that gave one identity, the end of safety. And at such a moment, unable to see and not daring to imagine what the future will now bring forth, one clings to what one knew; or thought one knew; to what one possessed or dreamed that one possessed. Yet it is only when man is able. without bitterness or self-pity, to surrender a dream he has long cherished, or a privilege he has long possessed, that he is set free--that he has set himself free--for higher dreams, for greater privileges [Ref. 10].

There are several reasons associated with the human need to hold on that Tannenbaum discusses in his book on the subject. Four specific points about holding on made by Tannenbaum are [Ref. 11]:

- Change is loss. All change requires letting go of the familiar and predictable. Emotions normally associated with loss include anger, guilt, grief, helplessness, hopelessness, and depression.

- Change is uncertainty. All change requires a move from the known to the unknown. Past experience with ambiguity and surprise will impact attitudes toward change. Emotions normally associated with uncertainty may be fear, panic, dread and anxiety.

- Change dissolves meaning. All change dissolves some past meaning in the lives of people experiencing it. That meaning may or may not be replaced. This is directly related to the self definition people receive from their jobs. Emotions normally associated with loss of meaning are confusion, anxiety, frustration, boredom, apathy and depression.

- Change violates scripts. Scripts are the deeply held plans and goals for our lives, conscious or perhaps unconscious, that form our approach towards life itself. Change can represent a loss of this script, or produce unacceptable variations on his script. Emotions normally associated with this loss are deep and powerful, such as shame, guilt, and rejection.

These aspects of change cause powerful emotional responses in the people facing change. It is natural for those people to want to hold on to the old ways, the things known. People want the predictability and security of the familiar. Even though the potential for improvement is being forecast with the change, people know their current needs are being met without the uncertainty. By holding on, people can avoid having to let go of practices that made them comfortable, and avoid the risks they feel are inherent with any change. The change doesn't have to be major to cause these reactions. Ask any manager who has rearranged a work space about people's reaction to that minor change.

3. Transition Versus Change

Some theorists in organizational development would take issue with the semantics of Hoffman's earlier quotation. Bridges would argue that the real fear is not of change, but of the transition itself. He feels that people are more open to the idea that change must occur, but it is the transition that is resisted. "Change often starts with a new beginning, but transition must start with an ending--with people letting go of old attitudes and behaviors [Ref. 12]."

The leader who recognizes the dangers of transition, and incorporates methods to answer those dangers in his plan will be the most likely to succeed. It is ironic that planned changes to the structure of an organization are carefully thought out, while organizational transitions are not. To manage transition properly, Bridges says the leader must [Ref. 12]:

- Identify the impacts of the planned change
- Identify the individuals most affected by the change

- Assess the units readiness for transition
- Analyze the political implications of the change
- Set a realistic pace for transition
- Create a team to manage the transition
- Identify new skills required and conduct training
- Review and improve communication resources of organization
- Create incentives for meeting the requirements of transition
- Plan to celebrate the different phases of transition

One of the specific tools Bridges uses to analyze the impact of the transition is the transition management plan. This involves forecasting the impact of the transition methodically, and using trained organizational development specialists to prepare the organization for the difficult transition ahead. If the resources are not available for the services of such a specialist, a dynamic leader can use the tools and guidelines that bridges proposes to help lead his organization through the transition state. Figure 9 depicts a sample tool from Bridges' book that the transition leader can use to evaluate the impact of loss on the organization.

What	**WHO**			
	Groups	**Individuals**	**Outsiders**	**You**
Turf				
Attachments				
Structure				
Future				
Meaning				
Control				
Identity				

Figure 9. Impact of Loss Matrix

By taking these steps, the leader can account for the effects of transition and help the people of the organization to better cope with the uncertainty that will certainly be produced in the transition state.

4. The Transition State

In planning for a transition, the leader should dedicate effort to considering the effects of the transition state [Ref. 13]:

- High uncertainty, low stability
- High levels of perceptual inconsistency
- High emotional stress on the people
- High energy (often undirected)
- Control becomes a major issue
- Conflict increases
- Past patterns of behavior become explicitly valued.

Because there are so many volatile emotions and feelings involved in managing this state, many people prefer to pretend it doesn't exist. It is much easier to dismiss the requirements of concerned leadership throughout the transition state as "touchy-feeley" management. Another term for the transition state is the "neutral zone". The failure of traditional management training to properly account for the neutral zone has handicapped the ability of the mangers to successfully navigate the treacherous gap between the new ways and the old [Ref. 12]. The plan for getting through transition must include two key areas. It must provide a strategy for supporting the people who are experiencing the emotional symptoms mentioned earlier, and it must also provide a mechanism to harness the creativity and energy that will be produced by the transition itself.

As Bridges states in his book, "If all of this sounds a little strange and unbusinesslike, that may simply show why so few businesses manage transition successfully." [Ref. 12] Since leaders are not trained in how to manage transition, the instinct is to get through it as quickly as possible, and not dwell on the feelings of people.

One of the painful, yet probable realities of transition is that there may be people who, no matter what steps are taken by the leaders of the change, will not be able to adapt. Leaders must be willing to acknowledge this, and transfer or termination of employment of these individuals may be best for them and the organization in transition. It is important to recognize these people as early as possible, because depending on their social capital in the organization, they may become leaders of resistance that might possibly prevent or sabotage the transition.

5. Training for Transition

One of the easiest, yet frequently missed steps in planning for transition is the step of training. The identification of new skill sets required by the transition will enable the planners to couple required training with the people who will be expected to perform new functions as a result of the change. This training will build the confidence of the people in transition, help them to see the benefits of the new system, and increase the likelihood of success. As Bridges says,"It is surprising how often organizations pay a great deal of money for new equipment and refuse to pay a little money to train people to use it properly [Ref.12]."

B. THE HUMAN FACTORS OF TRANSITION

While the primary focus of this thesis has centered on technology and design issues, it would have been incomplete without acknowledgment of the human factors which have shaped the evolution of some of the decisions that were made. As in any project which will change a person's environment, there has been some resistance to change on the part of people at MCI. While this is an expected outcome of the process, migration planners need to give this facet of the project due consideration.

1. The Psychology of Resistance to Change

While preparing this architectural recommendation, the project team was not yet educated in the psychology of resistance to change. Bryant has identified seven factors that influence attitudes towards change:

- Basic predisposition to change
- Personal sense of security
- Prevailing cultural beliefs
- Extent of trust and loyalty
- Objective historic events
- Specific apprehensions and expectations about the particular change [Ref. 14]

These key factors, often ingrained from a very early age, are the framework around the manner in which any person will deal with change. If these factors cannot be incorporated into new system planning, then the planners should not be surprised or disappointed when "their brilliant ideas and logical analysis are simply rejected [Ref. 14]."

2. Classes of Personalities With Respect to Change

It is now well documented in this chapter that many people dislike change. The uncertainties of transition bring with them many factors that merit special consideration in order to properly manage the transition effort. By addressing the type of change attitude held by key people, their attitudes can be incorporated into the transition plan and, hopefully, either converted to believers in the new system or at least neutralized in the damage they cause to the transition effort. According to Shaffer and Simon, people fall into five classes regarding change [Ref. 15]:

- Change positive - excited about change, welcome it in all forms. Very rare.

- Change neutral - no strong opinions, go with the flow, adapt to the organization. Not a change leader, but not an enemy either.

- Change reluctant but open minded - large category of people. May champion the status quo until it is obvious that a change is occurring. Often a good balance to change positive people because they are more skeptical. Can be convinced of the merits of new systems with proper groundwork.

- Change combative - Strong personalities in the organization. Typically have invested significant time in getting the organization to its current state. Quick to decry the merits of any change. It is very possible that these personalities will not be able to adapt to the changes required to progress.

- Change progressive but agenda driven - Very harmful. Appear to support the transition, but are guided by personal agendas not known to the planner.

By becoming familiar with these classes of attitudes, identifying where key people fit in to these classes, and preparing for ways to address the concerns and fears of these people, the planner can reduce the impact of change resistance on the migration to a new system.

C. TRANSITION AT MCI

With the theoretical foundation for understanding transition established, analysis of the applicability of that foundation to the MCI project is the next logical step. In retrospect, the human factors of this project were the most important. The inexperience of the project team in anticipating the impact of the human factors contributed significantly to the learning process for the team. In the sections that follow, some of the specific issues of change that impacted this study will be discussed.

1. "Holding On" to the Minicomputer

Much of the senior leadership of the MIS section of MCI has spent their entire career operating mainframes and minicomputers. For many people, the idea that microcomputers are now as powerful as the venerable minicomputer is very difficult to accept. The experience base of knowledge and personal credibility in the organization is tied directly to the organizational power that generates from the ability to operate the minicomputer. Transition to a microcomputer based system puts the senior leaders in the position of knowing no more, or perhaps even less than their subordinates. This potential loss of credibility is significant incentive to "hold on" to the old system architecture. In interviewing MIS personnel, many comments were made about the old system being good enough, how there was no need to change the minicomputer out, and how the operating system and development environment was fine for the need of the system. All of these rationalizations indicate a strong tendency to "hold on" to the old ways.

2. Loss of Turf

Closely related to the need to hold on is the loss of turf that is associated with giving up the sole control of a centralized computer system. The "turf" for the context of this discussion is the control of the system. If the system is transitioned to a microcomputer environment with a common network operating system, the networking section would be better suited to run the system: The MIS section would be left with no real duties. This is a loss of turf, and along with it, potential loss of job security.

3. Required Training for Transition

There are numerous opportunities for training to ease the burden of transition. Training for personnel in the selected database is available to ease the fear of moving to a new DBMS, and as Chapter VII will point out, if a planned migration is implemented, training in the end state operating system and hardware could begin immediately. The overall strategy of training is to reduce the fear of the new system by increasing staff familiarity with it.

4. Analysis of the Key Personnel Classes

The predominant personality class that was faced at MIS in developing a system architecture was the "Change progressive, but agenda driven personality type". From the outset of this project, MIS planners at MCI had intended to conduct a modest improvement in the existing database system, without addressing the rest of the architecture. Further study of the proposal and involvement of Training and Education (T&E) Division at Marine Corps Combat Development Command forced MCI to consider a more in depth analysis and design effort. To get funding to move forward with the project, MCI was forced to solicit involvement of the Naval Postgraduate School.

By the time NPS had formed a project team and had begun initial analysis of the project, the current state of customer service at MCI was receiving very high levels of attention from the Commandant of the Marine Corps. The time frame required by the NPS project team to conduct the analysis provided MCI with enough breathing room to conduct some modifications to customer service problems which reduced the level of command attention directed at MCI.

The level of cooperation received by the project team shifted dramatically late in 1996. It became obvious to the team that there were other agendas that were being addressed outside the redesign effort. Return of information that had been promised for September in order to meet a December deadline was not forthcoming until January, after the deadline had passed. Information necessary for the completion of phases of the project was delayed inordinately without justification. Major decisions affecting the architecture were made, and the project team was not included in the decision, nor briefed after the decision had been made. The agendas that were driving the actions of the key players in this process are not as salient as the fact that these agendas existed and were a major shaping force in the outcome of the project.

As a final note on the impact of transitional factors on this project, the NPS project team recommended option one, with a phased migration to option three. MCI chose to pursue option two as an architectural solution. The choice of this architecture is attributed to the hidden agendas of the key MIS decision makers, and is not recommended by the NPS team as the optimal solution.

D. RECOMMENDATIONS FOR MCI

There are many facets of transitional management to address. The recognition that any area of transitional management should be considered is to the credit of a migration planner. Key areas of transitional awareness that should be addressed for their impact on the migration effort include the relationships of key staff members, conflict management, internal communications, social structure and culture of the workplace, dominant coalitions between workers, rewards or incentives for transition, integrating mechanisms to facilitate the transition, worker input to the transition process, and worker attitudes toward change.

There are several steps that can be taken by the transition planner to address the human element of change. These include education, developing and obtaining the support of senior leadership, getting the users of the system involved in the development, rewarding those who support the change and those who change, using the change as a mechanism to support career enhancement for the system users.

The leader can complete transition training, or educate himself on organizational development techniques so that training in the expectations of transition itself can be conducted for the unit. Education in the mechanics and characteristics of humans that are undergoing change is a useful undertaking for the transition leader. By acknowledging the human factor in the success of a migration and transition effort, the project manager can begin to increase the success probability by managing the risk associated with the human element of the transition. MCI, like most military organizations, has a change resistant culture and many of the IS staff are change reluctant or change combative. MCI leadership can ease the transition to MCIAIS II and improve the likelihood of success by recognizing these human factors and working with the key change resistors.

VII. CONCLUSIONS AND RECOMMENDATIONS

The objective of this chapter is to summarize the research and results of this thesis, present conclusions, and suggest recommendations. It is organized into three sections. Section A provides a summary of the research effort. Section B examines the issues and conclusions raised through the use of the methodology, and Section C provides some recommendations for MCI that resulted from this study.

A. SUMMARY OF RESEARCH

As a part of a larger project, this study began in August 1996 in response to a request from the Marine Corps Institute to determine the feasibility and requirements for a new information system for MCI. The primary objective of this thesis is to develop a technology architecture to support the information systems of MCI and to address the complex technical and human resource issues of migration from the current legacy system to the new one.

To achieve the objective, this research addressed the requirements of a technology architecture for MCI, following Spewak's Enterprise Architecture Planning methodology. Additionally a plan for system migration was developed using the guidelines for incremental migration proposed by Brodie and Stonebraker.

B. ISSUES AND CONCLUSIONS

1. Research Questions Revisited

In Chapter I, the thesis raised several central research questions. Answers to these questions as a result of conducting this research are discussed below.

1. Can a technology architecture be developed to support the current and future needs of the Student Services Department at the Marine Corps Institute?

There is a number of technical solution options that could support the needs of SSD. These options were discussed in detail in Chapter IV. The recommended technology architecture is based on a detailed analysis of technology and the environment at MCI, and is summarized in the following section.

2. Can existing hardware and software used by the Student Services Department of the Marine Corps Institute be successfully migrated to an open system architecture?

The incremental strategy for migration proposed in Chapter V presents a plan that offers a migration strategy with the highest probability of success. If MCI adopts an incremental plan, thus resisting the desire to conduct "cold turkey" migration, it will significantly increase the likelihood of a successful transition.

3. Can Enterprise Architecture Planning support all the necessary requirements of this transition?

EAP methodology provides for the definition of a technology architecture within the definition of an overall system. As a methodology, it includes guidelines for implementing the plan and for managing the transition effort. It is weak, however, in the details of a migration strategy. For this reason, the incremental strategy developed by Brodie and Stonebraker was used to complement EAP in the area of migration.

4. What is the current state of the Marine Corps Institute Automated Information System (MCIAIS)?

The state of the current system was presented in detail in Chapter III. Appendix A also provides a complete summary listing of the hardware, software, and network components of the current system. As discussed, the system is barely meeting the current needs of MCI, and requires extraordinary maintenance efforts to meet even simple business changes. The lack of usable data models or process models compounds the problem of maintaining and evolving the system.

5. What combinations of hardware and software should be used to meet new system requirements within the given fiscal limitations?

There are many possible options of hardware and software combinations that could significantly improve the current state of the information system. Any selected option should support contemporary principles of developing hardware architectures as outlined. MCI can build upon the proposed data and process models to design and implement a contemporary information system that provides improved flexibility and customer support.

2. Research Issues

The methodology used for this study was designed to be used by people trained in the techniques of EAP, with the proper time and tools to undertake a major enterprise design effort. If conducted correctly, EAP appears to be a strong, methodical approach with which to address the development of a new information system. The detail and processes of EAP were impressive. It is a useful approach to system development and merits consideration bv anyone undertaking a system development project.

The problems of implementing EAP as a methodology became immediately apparent in the genesis of this study. The team chosen for the project was not trained in

EAP techniques. Because it was an academic undertaking, there was an expectation that a significant portion of the project itself would involve a learning curve on the part of the team members. Additionally, the scope of the study was below what was required for EAP. Spewak clearly states "Despite the best of intentions, and the intuitive appeal of dividing EAP along departmental lines and subsequently combining or integrating them, experience has shown that this does not happen successfully" [Ref. 1]. The scope of this study was clearly set below the enterprise level and therefore, by Spewak's own estimation, headed for trouble.

The third departure from the EAP methodology was the order of definition taken by the team. The database model was developed *before* the business functions were defined, not after the business process definition as proscribed by the methodology. The impact of this departure has not yet been evaluated.

Finally, the geographic separation and poor communications between the project team and MCI were problematic. The methodology requires a close working relationship with a high level of feedback and interaction between the project team and the supported enterprise. Competing priorities and physical separation made this essential requirement difficult.

C. RECOMMENDATIONS

The completion of this research culminates in specific recommendations for MCI with regard to the hardware, software, and migration to the a new system. Many factors were considered in the development of these recommendations. Standards were applied where required, industry trends were analyzed and interpreted for their applicability to this project, and the preferences of the MIS users at MCI were given heavy consideration. The impact of change management issues was also considered and the eventual recommendations include the impact of these issues.

1. Technology Recommendations

The principle concern in recommending a technology architecture is its likelihood of success/risk of failure. A microcomputer based hardware platform operating Windows NT (as represented by the third option discussed in Chapter IV) best conforms with the developmental principles guiding this study. It is therefore the preferred choice for long term success, but the current capabilities and training requirements of the personnel who will be required to implement the recommended system cannot be overlooked. Additionally, a microcomputer based option represents the greatest change for the people of MCI and therefore possesses an increased risk of failure if implemented at this time.

It is therefore recommended that MCI adopt a phased approach to upgrade the current technology platform. The first phase is to upgrade the current HP 3000 minicomputer and using it as a server, installing a new Oracle® DBMS under the current MPE/iX operating system, and replacing the current dumb terminals with Pentium level microcomputer clients. The architecture of this option adheres to the majority of developmental principles guiding this study and will support all current and near term system requirements for MCI. The second phase, which culminates in the implementation of the third option, incrementally migrates MCIAIS to a truly open system architecture and consists of an Intel-based server with multiple processors, Windows NT operating system, Oracle® DBMS, and state-of-the-art microcomputer clients.

There has been much discussion in the Marine Corps regarding the transition of Marine Corps network operating systems from Banyan Vines to Windows NT. If this transition occurs, MCI will be able to standardize operating systems on Windows NT and eliminate the high overhead cost in training required to operate the proprietary Hewlett-Packard operating systems. Standardized training in the use of common microcomputers versus diverse computer systems is an important benefit of this option.

2. Migration

a. Continued Migration

One of the keys to a useful information system is its flexibility in adapting to new business requirements and conditions. Information system planners must realize that information systems can no longer be viewed as static. IS systems must be continually evaluated for additional migration. It is not a matter of *if* continued migration is necessary, but only when it is necessary and in what form the migration will take.

The migration from the first phase system to the second phase system will be much simpler than the initial legacy migration because it will involve a modular design, the same DBMS and open architectures. By incrementally migrating modules to the client side where possible, further reductions in server processing loads will be achieved. By breaking out of the mainframe paradigm that has held MCI in place for many years, MCI can design and develop a system that will be responsive to future requirements instead of addressing past technology and current requirements.

Due to a reevaluation of the implementation timeline by MCI, the potential exists to adopt this interim, incremental migration strategy to the ultimate recommended architecture. With the implementation of the gateway products discussed below, and minor hardware upgrades, it is believed that MCI can accomplish all of the requirements to begin a satisfactory transition to the second phase of MCIAIS. We strongly believe that this option is the most cost effective, training effective and consistent alternative to meet the stated objectives of this study. The migration to a relational database on the current platform is a natural incremental step to the end state system. This phased approach to system migration is integral to a successful transition for MCI.

b. Gateways

During the evolution of this project, only one potential commercial solution has been found for the migration of data directly from a TurboImage database to an Oracle® database, addressing the current needs of MCI. This potential solution, available from Starvision Inc., is a gateway application that provides an out of box gateway solution for TurboImage to Oracle® translation. There are two variants: the unidirectional gateway and the bi-directional gateway. The unidirectional gateway allows users of the TurboImage applications to update the Oracle® database or allows users of the new GUI windows based applications to update the TurboImage database, but not both. A bi-directional gateway is required to get a bi-directional functionality. This product provides full replication of data and best supports the parallel migration plan in its current form. The Starvision product was developed internally for Hewlett-Packard and has been improved and marketed separately from the original HP product line. It specializes in MPE/iX TurboImage gateway capability, and seems to be the most logical fit for the existing requirements at MCI.

Additional advantages of this product on a HP 3000 include significantly enhanced system performance and throughput of data due to “Standby Agents”. These are standby processes that are running before the actual requests are made by the user. This results in predefined data views being available much faster. The Starvision gateway is compatible with most development tools chosen for client side development, and is fully integratable into an intranet environment due to its open system design. This solution, or any other gateway that will support this type of gateway requirement, is the type of middleware needed to adopt an incremental migration from the current database to an Oracle® database solution.

c. Human Resources

To address the human element of change there are several steps that can be taken by the transition planner. These include education, developing and obtaining the support of senior leadership, getting the users of the system involved in the development, rewarding those who support the change and those who change, using the change as a mechanism to support career enhancement for the system users, and educating the organization in the characteristics of transition. Key areas of transitional awareness that should be addressed for their impact on the migration effort include the relationships of key staff members, conflict management, internal communications, social structure and culture of the workplace, dominant coalitions between workers, rewards or incentives for transition, integrating mechanisms to facilitate the transition, worker input to the transition process, and worker attitudes toward change.

MCI, like most military organizations, has a change resistant culture and many personnel are resistant to change. By recognizing these human factors and working with the key change resistors, MCI leadership can ease the transition to the next phase of MCIAIS and improve its likelihood of success.

APPENDIX A - INFORMATION RESOURCE CATALOG

In each of the following sections, the first part of the section will contain a listing of the information format for that section. The remaining parts of that section will reference the format provided for that section.

A. Applications

1. Format

The following format for the catalog of applications is used:

a. Name of application, and person primarily responsible for app.

b. Plain language definition of the application and what it does

c. Whether application is batch, on-line or both

d. Time frame required to run application

e. Frequency of application run

f. Applications/software that must be run before other applications can run

g. Applications/software that must be run after other applications have been run

2. Application One

a. Biscom Facsimile Service, SSgt Broome

b. Receives faxes electronically and forwards them to the final recipient based on 4-digit DMTF routing number (if included by sender of fax). Sends electronic faxes originating from users' personal computer.

c. On-line.

d. Sub-second.

e. As required.

f. None.

g. None.

3. Application Two

a. Source Library System (SLS), SSgt Broome

b. Stores electronic versions of Unit Activity Reports. When a unit desires to receive their UAR electronically, they send an e-mail request to the SLS for the specific UAR they want. The SLS sends them an e-mail with the requested UAR as an attachment.

c. On-line.

d. Sub-second.

e. As required.

f. None.

g. None.

4. Application Three

a. Plato Self-Paced Courseware, SSgt Broome.

b. Plato is a personal education interactive software package that is available on MCI's classroom computers. This software is used to provide training/education in several areas (English, Math, History, etc.) and is used by Course Developers to polish their writing skills and by Marines to improve their general knowledge and prepare for ASVAB tests.

c. On-line CD-ROM (CD-ROM server).

d. Sub-second.

e. As required.

f. None.

g. None.

5. Application Four

a. Lotus SmartSuite Self-Paced Courseware, SSgt Broome.

b. Interactive courseware installed on one classroom computer to teach MCI personnel how to use Lotus SmartSuite software.

c. On-line.

d. Sub-second.

e. As required.

f. None.

g. None.

6. Application Five

a. Shark!mail, SSgt Broome.

b. Windows-based e-mail handling package that includes messaging rules for VINES. Run by users from LAN server.

c. On-line.

d. Sub-second.

e. As required.

f. None.

g. None

7. **Application Six**

   a. NetCensus, ver 2.80, SSgt Broome

   b. Automated LAN hardware/software inventory system. Periodically, when a user logs on to the LAN, his PC's hardware and software components are automatically inventoried by NetCensus. NetCensus resides on the LAN server and is activated at an interval determined by LAN Administrator (ISMO).

   c. On-line

   d. N/A (no user interaction)

   e. Periodically (usually once/week)

   f. None

   g. None

8. **Application Seven**

   a. Lotus SmartSuite (office integrated suite), SSgt Broome.

   b. Personal productivity software.

   c. On-line.

   d. Sub-second.

   e. As required.

   f. None.

   g. None.

9. Application Eight

a. Calendar Creator + (calendaring), SSgt Broome.

b. Personal productivity software.

c. On-line.

d. Sub-second.

e. As required.

f. None.

g. None.

10. Application Nine

a. Microsoft Project (project management), SSgt Broome.

b. Personal productivity software.

c. On-line.

d. Sub-second.

e. As required.

f. None.

g. None.

11. Application Ten

a. Fastrack (project management), SSgt Broome.

b. Personal productivity software.

c. On-line

d. Sub-second.

e. As required.

f. None.

g. None.

12. Application Eleven

a. IBM Anti-Virus (anti-virus), SSgt Broome.

b. Personal productivity software.

c. On-line.

d. Sub-second.

e. As required.

f. None.

g. None.

13. Application Twelve

a. Delrina Form Flow (automated forms), SSgt Broome.

b. Personal productivity software.

c. On-line.

d. Sub-second.

e. As required.

f. None.

g. None.

14. Application Thirteen

a. Netscape (Internet access), SSgt Broome.

b. Personal productivity software.

c. On-line.

d. Sub-second.

e. As required.

f. None.

g. None.

15. Application Fourteen

a. PKZIP (file compression), SSgt Broome.

b. Personal productivity software.

c. On-line.

d. Sub-second.

e. As required.

f. None

g. None.

16. Application Fifteen

a. ABC Flowcharter (flowcharting), SSgt Broome.

b. Personal productivity software.

c. On-line.

d. Sub-second.

e. As required.

f. None.

g. None.

17. Application Sixteen

a. MTF Editor (naval messages), SSgt Broome.

b. Personal productivity software.

c. On-line.

d. Sub-second.

e. As required.

f. None.

g. None.

18. Application Seventeen

a. Reflections (Hewlett-Packard terminal emulation), SSgt Broome.

b. Personal productivity software.

c. On-line.

d. Sub-second.

e. As required.

f. None.

g. None.

19. Application Eighteen

a. Adobe PhotoShop (graphics/desktop publishing), SSgt Broome.

b. Personal productivity software.

c. On-line.

d. Sub-second.

e. As required.

f. None.

g. None.

20. Application Nineteen

a. CorelDraw (graphics), SSgt Broome.

b. Personal productivity software.

c. On-line.

d. Sub-second.

e. As required.

f. None.

g. None.

21. Application Twenty

a. VEFADDR and RUCMCC Upload, GySgt Floyd.

b. Retrieves most recent student information (SSN, rank, location, etc.) in ASCII format from the Marine Corps Total Force System (MCTFS) and refreshes MCIAIS student database.

c. Batch

d. Part 1: 7 hours, Part 2: 30 mins

e. Every third nightly cycle.

f. This job can be run independently of the nightly cycle jobs remaining below. It is performed in two parts: the first part transferring the dataset containing the information from MCTFS to MCI's Hewlett-Packard over a leased line circuit, and the second part overlaying the MCTFS information in MCIAIS. The first part is performed during the day, and the second part is performed during the nightly cycle.

22. Application Twenty-one

a. JDLYBKUP.JOB.MCIAIS, GySgt Floyd.

b. Makes copies of all input files. [Note: The MPE/ix operating system does not support generation datasets; therefore, it is not easy to keep several months' worth of input information for research purposes. We could do this programmatically, but frankly have not done so because of the significant effort involved. A desired feature in the "new MCIAIS" would be the ability to store input data for at least the past two months to allow us to research problem transactions as necessary.]

c. Batch

d. 1 min

e. Nightly

f. First job of nightly cycle

g. Must run before JDAILY01.JOB.MCIAIS

23. Application Twenty-two

a. JDAILY01.JOB.MCIAIS, GySgt Floyd.

b. Purges previous night's enrollment, status, and holdfile batch files and repopulates them with today's input. Processes enrollment, reenrollment, and administrative deletion transactions. Archives address file. Produces

Enrollment Error Report and Deputy Director Report (sends report datafiles to the print spool to be printed later by system operator).

c. Batch

d. 2 -10 mins, depending on amount of input

e. Nightly

f. Must run after JDLYBKUP.JOB. MCIAIS

g. Must run before JDAILY2A.JOB.MCIAIS (Mondays only) or JDAILY2B.JOB.MCIAIS (Tues-Fri)

24. Application Twenty-three

a. JDAILY2A.JOB.MCIAIS, GySgt Floyd

b. Runs the holdfile program, grading program, and motivations and disenrollments program. Updates onhand quantities stored in the Logistics AIS database (sub-database of MCIAIS). Produces the Total Enrollments report and Holdfile report (sends report datafiles to the print spool to be printed later by system operator).

c. Batch

d. 7 mins

e. Mondays

f. Must run after JDAILY01.JOB. MCIAIS

g. Must run before JDAILY03.JOB.MCIAIS

25. Application Twenty-four

a. JDAILY2B.JOB.MCIAIS, GySgt Floyd

b. Runs the holdfile program and grading program. Updates onhand quantities stored in the Logistics AIS database (sub-database of MCIAIS). Produces the Total Enrollments report and Holdfile report (sends report datafiles to the print spool to be printed later by system operator).

c. Batch

d. 5 mins

e. Tuesdays through Fridays

f. Must run after JDAILY01.JOB. MCIAIS

g. Must run before JDAILY03.JOB.MCIAIS

26. Application Twenty-five

a. JDAILY03.JOB.MCIAIS, GySgt Floyd

b. Creates mailing labels (sends label datafile to the print spool to be printed later by system operator).

c. Batch

d. 2 - 10 mins

e. Nightly

f. Must run after JDAILY2A.JOB. MCIAIS or JDAILY2B.JOB.MCIAIS

g. Must run before JDAILY04.JOB.MCIAIS.

27. Application Twenty-six

a. JDAILY04.JOB.MCIAIS, GySgt Floyd

b. Creates and prints student status cards (R-6 Cards), letters notifying students of failed exam (R-69), completion certificates and letters detailing questions missed and their correct answers (R-69), and mailing labels for PME program diplomas. Creates Daily Stock Status Report, Logistics Required Action Report, Logistics Transaction Report, and PME Completion Report. Creates file with PME diploma information to be printed separately by SSD on HP Laserjet printer. (Sends report, label, and status datafiles to the print spool to be printed later by system operator.)

c. Batch

d. 6 - 10 mins

e. Nightly

f. Must run after JDAILY03.JOB. MCIAIS

g. Must run before JDAILY05.JOB.MCIAIS

28. Application Twenty-seven

a. JDAILY05.JOB.MCIAIS, GySgt Floyd

b. Creates datafile containing enrollment, completion, and disenrollment transactions to be posted to the MCTFS ("D91" file). Processes MCTFS Advisory file errors (the Advisory file contains MCI transactions that failed to post in the previous night's cycle).

c. Batch

d. 1 - 10 mins

e. Nightly

f. Must run after JDAILY04.JOB. MCIAIS

g. Must run before JDAILY06.JOB.MCIAIS

29. Application Twenty-eight

a. JDAILY06.JOB.MCIAIS, GySgt Floyd

b. Counts exams issued during cycle. Creates Exam Total report (sends report datafile to the print spool to be printed later by system operator)

c. Batch

d. 1 - 5 mins

e. Nightly

f. Must run after JDAILY04.JOB. MCIAIS

g. Must run before PARTBKUP.JOB.MCIAIS

30. Application Twenty-nine

a. PARTBKUP.JOB.MCIAIS

b. Performs partial backup of MCIAIS datafiles, program files, and databases, updating any changes made since the last backup of any kind.

c. Batch

d. 60 - 90 mins

e. Nightly (except Fridays and Monthlies, when full backup is performed)

f. Last job in cycle

31. Application Thirty

a. J3633UA4.JOB.MCIAIS, GySgt Floyd

b. Creates datafiles containing Unit Activity Report (UAR) data such as RUC, SSN, name, etc. (These files will be copied onto tapes in jobs FCOPYUA1.JOB.MCIAIS through FCOPYUA3.JOB.MCIAIS, which will be downloaded to print on MCI's Xerox printer.)

c. Batch

d. 8 hrs

e. Monthly

f. Must run after JDAILY06JOB. MCIAIS

g. Must run before J3633UB4.JOB.MCIAIS

32. Application Thirty-one

a. J3633UB4.JOB.MCIAIS, GySgt Floyd

b. Creates Monthly Report of Operations, Program Enrollments and Completion report, Answer print, Reference print, Air Force Completion report, Exam Totals report Logistics AMRD Variance report, Course Data File Print, Command and Staff Course report, and RUCMCC Listing (sends report datafiles to the print spool to be printed later by system operator).

c. Batch

d. 10 - 14 hrs

e. Monthly

f. Must run after J3633UA4.JOB.MCIAIS

g. Must run before J3633UC4.JOB.MCIAIS

33. Application Thirty-two

a. J3633UC4.JOB.MCIAIS, GySgt Floyd

b. Archives course records whose date of latest course activity is greater than 13 months.

c. Batch

d. 30 - 60 mins

e. Monthly

f. Must run after J3633UB4.JOB.MCIAIS

g. Must run before FULLBACK.JOB.MCIAIS

34. Application Thirty-three

a. J3633UC4.JOB.MCIAIS, GySgt Floyd

b. Archives course records whose date of latest course activity is greater than 13 months.

c. Batch

d. 30 - 60 mins

e. Monthly

f. Must run after J3633UB4.JOB.MCIAIS

g. Must run before FULLBACK.JOB.MCIAIS

35. Application Thirty-four

a. FULLBACK.JOB.MCIAIS, GySgt Floyd

b. Performs a full backup on entire MCIAIS (datafiles, program files, and databases)

c. Batch

d. 10 hours

e. Fridays and monthly

f. Can run after JDAILY06.JOB.MCIAIS or J3633UC4.JOB.MCIAIS

g. Must run before Electronic UAR Process (JUSLS.JOB.MCIAIS)

36. Application Thirty-five

a. JUSLS.JOB.MCIAIS, GySgt Floyd

b. Sets up file framework for electronic UARs.

c. Batch

d. 20 mins

e. Twice monthly

f. Can run after FULLBACK.JOB.MCIAIS, but is independent of the monthly closeout cycle process.

g. Must run before J3633OLU.JOB.MCIAIS

37. Application Thirty-six

a. J3633OLU.JOB.MCIAIS, GySgt Floyd

b. Populates the electronic UAR framework set up in JUSLS.JOB.MCIAIS with student/course information.

c. Batch

d. 8 - 10 hrs

e. Twice monthly

f. Must run after JUSLSL.JOB.MCIAIS

g. Last job in electronic UAR creation process. The datafile created in this job is then moved from the HP to one of the LAN servers by the system operator.

38. Application Thirty-seven

a. FCOPYUA1.JOB.MCIAIS, GySgt Floyd

b. Copies UAR datafiles created in J3633UA4.JOB.MCIAIS to Tape 1 (approximately 500,000 student records).

c. Batch

d. 20 mins

e. Monthly

f. May run after J3633UA4.JOB.MCIAIS

g. Must run before FCOPYUA2.JOB.MCIAIS

39. Application Thirty-eight

a. FCOPYUA2.JOB.MCIAIS, GySgt Floyd

b. Copies UAR datafiles created in J3633UA4.JOB.MCIAIS to Tape 2 (approximately 500,000 student records).

c. Batch

d. 20 mins

e. Monthly

f. Must run after FCOPYUA1.JOB.MCIAIS

g. Must run before FCOPYUA3.JOB.MCIAIS

40. Application Thirty-nine

a. FCOPYUA3.JOB.MCIAIS, GySgt Floyd

b. Copies UAR datafiles created in J3633UA4.JOB.MCIAIS to Tape 3 (approximately 500,000 student records).

c. Batch

d. 20 min

e. Monthly

f. Must run after FCOPYUA1.JOB.MCIAIS

g. Last job in monthly cycle.

41. Application Forty

a. PART 1 (JBAGTFLE.JOB.MCIAIS), GySgt Floyd

b. Job JBAGTFLE.JOB.MCIAIS creates ASCII file on HP. System operator then FTPs it to Conversant over the LAN.

c. Batch

d. 3 -4 hrs

e. Once per week

f. First job in process

g. Must run before processes in Part 2 (below) are started.

42. Application forty-one

a. PART 2 (Conversant Load), GySgt Floyd

b. System operator logs onto Conversant system and downloads datafile that was FTP-ed in Part 1 (above) onto Conversant database.

c. Batch

d. 30 mins

e. Once per week

f. Must run after Part 1 (above)

g. Last job in Conversant database update process

B. HARDWARE

This section contains a brief description of major hardware items in the current information system.

1. Servers

Three Dell PCs located at MCI on the Washington Navy Yard (two are 486-66s w/32MB memory and 7.5GB disk arrays; one is a 80586-60MHz w/64MB memory and 8GB disk array). Two Dell PCs located at the Marine Barracks Washington DC (one is a 486-66 w/32MB memory and 7.5GB disk array; one is a 80586-60MHz w/64MB memory and 8GB disk array). These servers contain file, print, and e-mail services for MCI and Post users. All servers run Banyan VINES version 6.3(0). Each is Single Attached Station (SAS)-connected to a FDDI backbone for server-to-server traffic, and has a separate 10MBS ethernet connection for user traffic. The topology is Ethernet, using Cat 5 UTP cabling, with remainder using thin ethernet cabling (to be replaced with Cat 5 UTP cabling by Apr 97).

2. Network communication platforms

Two Cableton MMAC-8 Hubs, with one management module (w/SNMP) per hub, four 24-port 10baseT ethernet modules for user ports per hub, and one FDDI module per hub (four SAS ports, one DAS port -- the three LAN servers located at MCI are each attached to the FDDI module via one of the SAS ports).

One AirLan/Bridge, with Antenna, Long Range, 11db directional wireless connection between MCI Server 1 and MarBks Server 1 (2 MBS data transfer rate, spread-spectrum transmission, 900 MHz bandwidth, rapid switching between channels).

3. Microcomputers

There is a mixture of 80386, 80486, and 80586. By mid-1997, most microcomputer platforms at MCI will be 80586-90MHz+ with 16-40MB RAM, 1.2+GB hard drives, Windows 95. No 80386-based micros will remain in service. There will be (18) 80586s in SSD, and (7) 80586s in Logs. MCI prefers to keep Win95 on user ("client") microcomputers, and use WindowsNT as the operating system for application servers -- or users running high-end PC applications.

4. Minicomputer

Hewlett-Packard 3000, Series 957SX (HP Prod No. 32651B) w/160MB memory

5. Input Devices

a. Scanners: One 360 DPI, One 600 DPI

b. Digital Camera: One Kodak Digital Camera System

c. Tape Drive: 6250/1600 cpi Streaming Tape Drive (HP Prod No. 7978B)

d. Scanner: National Computing Systems, model Op7-35, Dual Ink Scanner. Exam answers are scanned from DP-37 input forms onto diskette, which are then read into the nightly cycle and processed by the MCIAIS grading application.

e. Hewlett-Packard Terminals or PC Terminal Emulation (HP Prod No. 2392A)

6. Output & Graphic Displays (printers)

a. Hewlett-Packard LaserJet 3 (personal printer) -- total MCI quantity: 9 (1 in SSD, 2 in Logs)

b. Hewlett-Packard LaserJet 3si (personal printer) -- total MCI quantity: 1 (1 in Logs)

c. Hewlett-Packard LaserJet 4 (personal printer) -- total MCI quantity: 20 (1 in SSD, 0 in Logs)

d. Hewlett-Packard Laserjet 4si (network printers) -- total MCI quantity: 5 (1 in SSD, 0 in Logs)

e. Hewlett-Packard Line Printer (One) 1600 LPM Line Impact Printer (HP Prod No. 2567C)

f. Xerox Line Printer (One) Model 4850 Laser Printer (prints black and one other color / 200,000 impressions/mo capacity)

7. Plotters

None

8. Storage Media (disks, cd's, tape, etc.)

a. CD-Rom Tower with capability for six CD-ROMs and one rewriteable CD-optical disk for off-line storage of MCI course materials. CD-ROMs are now 2x, but are planned to be upgraded to 8x in mid-97.

b. (One) Series 6000 SCSI Mass Storage System (HP Prod No. C3023R), which contains (four) 2GB disk drives.

c. (Two) 2GB full height SE SCSI disks (HP Prod No. A2446A)

d. (One) 2GB DDS DAT Drive (HP Prod No. C2477SZ)

9. Automated Voice Response (AVR) System

AT&T Intuity Conversant System, version 5.0. Conversant is an AVR system that connects to part of MCI's telephone PBX system (AT&T's Merlin Legend IS-3) [There are two Merlin Legends in MCI's PBX system -- one is a Merlin Legend IS-3 and the other is a Merlin Legend Intuity]. Callers to MCI are greeted by a recorded voice from the Merlin PBX and are offered several choices -- one of them being to access the Conversant AVR system to listen to a readout of student activity in MCI courses. If they desire to use the AVR system, they are prompted to enter the student's SSN and the course number for which they're interested in receiving status.

MCI will soon expand the capabilities of this system so that it will allow up to 12 callers to access it at a time (vice the current 6). Additionally, we desire to use Conversant as another method for Marine students to enroll in MCI courses.

Currently, Conversant resides on an AT&T-proprietary hardware platform (80386-based MAP/100C platform, 1.2GB hard drive, 32M RAM), running under UNIX System 5, version 4.0.

There is an Oracle 6 relational database installed on this platform, but it was never intended to handle the amount of student data we're loading into it. The database takes so much disk space that basic Conversant management programs have been removed to make room for this data. To restore these management programs, MCI is standing up a standalone PC (80586, Windows NT O/S, Oracle7 RDB) that will have the capacity to handle just MCI's student data in its database. Conversant will then access student data on this standalone database upon caller demand. This will free up the necessary space in Conversant to restore its original functionality (Conversant was never designed to hit against a relational database installed on its own hard disk). Conversant is connected to the HP CPU via thin ethernet adapters and thick ethernet to two MMAC-8 hub ethernet module ports. The purpose of this connection is to transfer student data from the HP to the Conversant student database. This update occurs once per week. The Merlin PBX is not connected to the HP 3000.

10. Planned hardware additions

a. 24-port 100MBS switched fast ethernet hub for application server traffic and high-speed connections for data processing personnel.

b. Remote Access Service (RAS) with PPP dial-in through Cisco router running on Windows NT platform.

C. SOFTWARE

1. Operating Systems

a. HP's MPE/ix (HP Prod No: 32651B), version 5.5, 60-user license

b. DOS 6.22

c. Windows 3.11

d. Windows 95

e. Windows NT version 4.0

2. Database Management Systems

a. HP's TurboImage/XL (prod no: B3524A), Version 4.0, 60-user license

b. HP's System Dictionary (Prod No. 32256A) -- data dictionary

c. HP's Query/V -- basic query application

d. Adager Corp's Adapter/Manager -- third-party DBMS tool (more powerful than HP's database utilities)

e. Cognos Corporation's Powerhouse Dictionary Language (PDL), ver 7.29.c -- creates and maintains Powerhouse data dictionary.

3. Languages

a. HP's Transact/iX (Prod No. 30138A) -- 3GL application development language

b. Cognos Corporation's Powerhouse 4GL, ver 7.29.c -- 4GL application development language

c. Cognos Corporation's Quiz, ver 7.29.c -- report and query writer

d. Cognos Corporation's QDesign, ver 7.29.c -- designs screens for data entry and retrieval

e. Cognos Corporation's QTP, ver 7.29.c -- volume transaction processor (can add, change, or delete large amounts of data quickly and efficiently)

4. Other Software

a. HP's Edit 3000-- line editor that creates and manipulates ASCII files

b. HP's TDP/3000 (Prod No. 36578A) -- full-screen editor

c. HP's GlancePlus MPE/iX (Prod No. B1787B) -- performance and tuning tool (installed with version 5.0 of MPE/iX O/S in winter '95)

d. Performance Software Group's Fastran -- compiler for Transact application development language

e. Performance Software Group's Facade, ver 1.1 -- full-screen co-editor

f. Dynamic Info Systems Corporation's Omnidex -- 3rd party indexing

g. Orbit Software's Backup+/XL -- high speed backup utility

h. Lund Performance Solutions' SOS Performance Advisor for HP3000 (SOS/3000) -- performance and tuning

i. Lund Performance Solutions' DefragX -- disc defragmentation package

j. Diamond Optimum Systems DOC 3000 -- scans and cross-references source code and job streams.

k. VE Soft's MPE/iX 3000 -- operating system utility that enhances system management, program development, and console operation tasks.

l. VESoft's Security 3000 -- system security package

m. VE Soft's VE Audit 3000 -- security reporting package (attempts to find loopholes in Security 3000 setup)

D. COMMUNICATIONS

1. Signal Devices

a. 28.8KBS Modems-- Currently, five dial-in phone lines are supported by MCI servers 1 and 2. Three of these five lines will be converted to PPP connections via a Windows NT Remote Access Service, to be attached to the hub via 100MBS ethernet cabling

b. HP's SupportLink (HP50759B) -- 2400 dial-in modem (being replaced with higher-speed modem)

c. Motorola CODEX 3500 -- modem/line conditioner, connects to 56KB circuit

2. Connections

a. From MarBks/MCI LAN to USMC WAN/MCDN: leased 56KB circuit attached between MarBks/MCI server 1 and HQMC server. This circuit will be upgraded to a T1 attached to a Cisco 4500M router, which will be SAS-attached to the MarBks/MCI LAN server-to-server FDDI backbone. Note: all TCP/IP traffic to the Internet shares this circuit with Banyan VINES traffic.

b. From MCI to KCMO: 56KB dedicated leased line (protocol is HP SNA)

c. From internal LAN PC users to HP: Ethernet LAN, "Reflections for Windows" terminal emulation software (protocol: VT Manager), soon to be Telnet-capable.

d. From HP to Xerox 4850 Printer: TCP/IP connection, using PC running interface software (interface software manufacturer: Solimar).

e. From HP to AT&T's Conversant (Automated Voice Response System): thin ethernet adapters and thick ethernet to two MMAC-8 hub ethernet module ports. Planned upgrade to 100BaseT cabling.

3. Network communications

a. (One) DTC72MX Communication Controller (HP Prod No. J2070A)

b. Hewlett-Packard's NS3000 (Prod No. 36920A) -- allows HP 3000 to communicate with other computers as part of distributed network. MCI uses the following modules:

c. SNA Link/ix (Prod No. 30291A): 3270 connections with IBM mainframe

d. SNA NRJE (Prod No. 30292A): print services, RJE

e. HP Telnet/iX : telnet capability (planned, not in use yet)

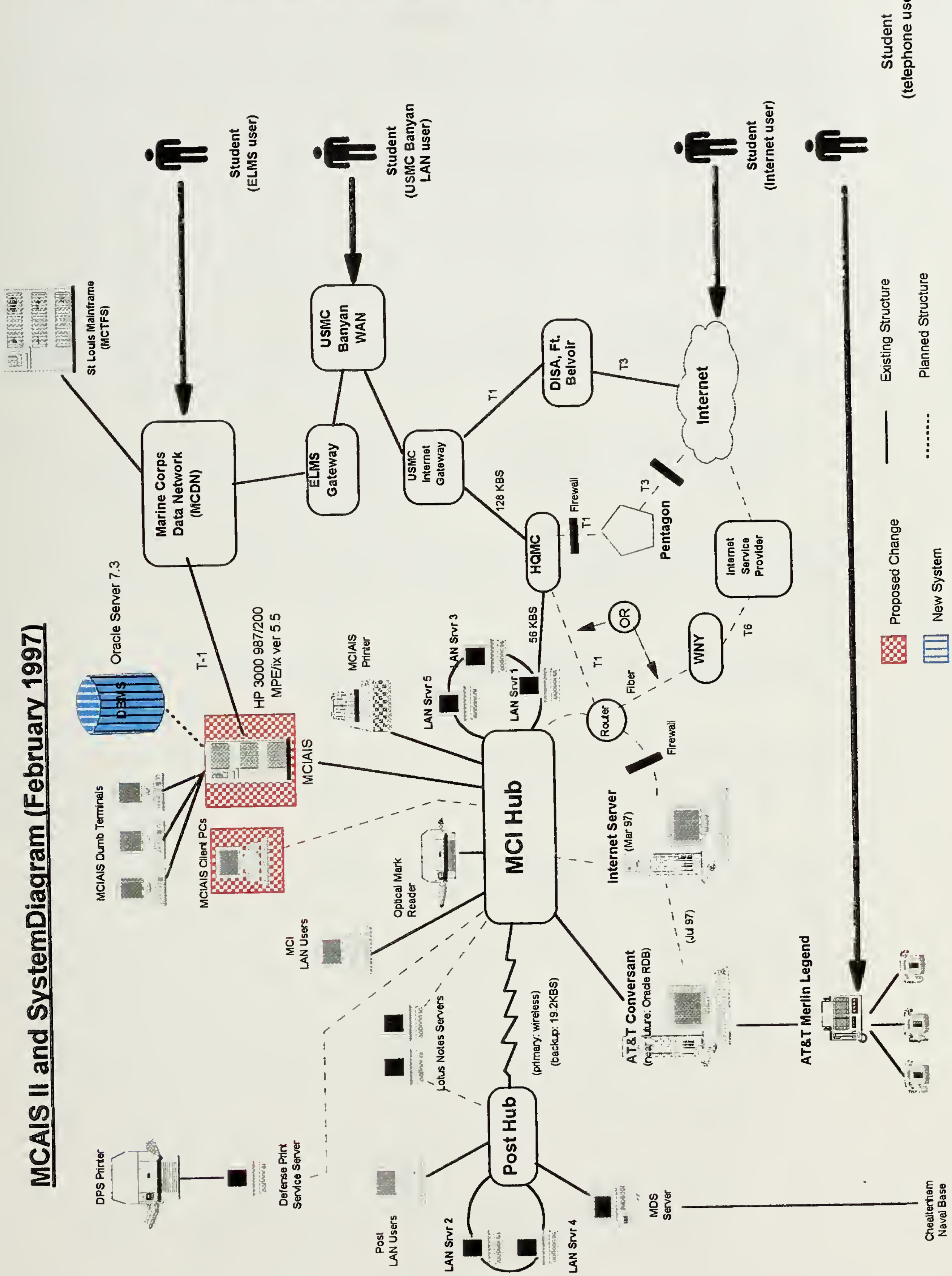
f. SNA IMF Pass Through: passes data fm HP to IBM mainframe

g. LU6.2 API: 3270 emulation through HP

h. SNA IMF/iX (Prod No. 30293A)

i. ThinLAN 3000/iX (36923A): connection into Banyan LAN

APPENDIX B- PROPOSED SYSTEM MAP



LIST OF REFERENCES

1. Spewak, Steven H., *Enterprise Architecture Planning,* John Wiley and Sons, Inc., New York, New York. 1995.

2. Zachman, John., *A Framework for Information Systems Architecture,* IBM Systems Journal, Vol. 26, No. 3, 1987.

3. Office of the Secretary of Defense, DOD Joint Technical Architecture, Version 1.0, Department of Defense, 1996.

4. Kamel, Magdi, et al, *Analysis, Design, and Prototype Implementation of a Contemporary Information System for the Marine Corps Institute, Final Report,* NPS-SM-97-006, NPS, 1997.

5. Brodie, Michael L. and Stonebraker, Michael. *Migrating Legacy Systems: Gateways, Interfaces, and the Incremental Approach.* Morgan Kaufman Publishers, Inc. San Francisco, California, 1995.

6. E-mail from brodie@gte.com (Michael L. Brodie), co-author of *Migrating Legacy Systems: Gateways, Interfaces, and the Incremental Approach,* dated April 09, 1997.

7. President and Fellows of Harvard College, *Leading Change,* Harvard Business School, Boston, Massachusetts, 1988.

8. Hoffman, Gerald M., *The Technology Payoff: How to Profit with Empowered Workers in the Information Age,* Irwin Professional Publishing, Burr Ridge, Illinois, 1994.

9. Nicholson, Nigel, *Career Systems in Crisis: Change and Opportunity in the Information Age,* Academy of Management Executive, Vol. 10, No. 4, 1996.

10. Baldwin, James, *Nobody Knows My Name,* Dial Press, New York, New York, 1961.

11. Tannenbaum Robert, and Hanna, Robert, *Human Systems Development,* Jossey-Bass Publishers, San Francisco, California,1985.

12. Bridges, William, *Surviving Corporate Transition,* William Bridges and Associates, Mill Valley, California, 1993.

13. Becknard , Richard and Harris, Rueben, *Organization Transitions,* Addison-Wesley Publishing Co., Reading, Massachusetts, 1987.

14. Bryant, Don., *The Psychology of Resistance to Change,* Management Services, March 1979.

15. Shaffer, Steven L. and Simon, Alan R. *Transitioning to Open Systems: Concepts, Methods, and Architecture.* Morgan Kaufman Publishers, Inc. San Francisco, California. 1995.

BIBLIOGRAPHY

1. Baldwin, James, *Nobody Knows My Name,* Dial Press, New York, New York, 1961.

2. Becknard , Richard and Harris, Rueben, *Organization Transitions*, Addison-Wesley Publishing Co., Reading, Massachusetts, 1987.

3. Bridges, William, *Surviving Corporate Transition,* William Bridges and Associates, Mill Valley, California, 1993.

4. Brodie, Michael L. and Stonebraker, Michael. *Migrating Legacy Systems: Gateways, Interfaces, and the Incremental Approach.* Morgan Kaufman Publishers, Inc. San Francisco, California, 1995.

5. Bryant, Don., *The Psychology of Resistance to Change,* Management Services, March 1979.

6. Hoffman, Gerald M., *The Technology Payoff: How to Profit with Empowered Workers in the Information Age,* Irwin Professional Publishing, Burr Ridge, Illinois, 1994.

7. Kendall, Kenneth E. and Kendall, Julie E., *Systems Analysis and Design, Third Edition,* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1995.

8. Kroenke, David M. *Database Processing, fifth edition.* Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1995.

9. President and Fellows of Harvard College, *Leading Change,* Harvard Business School, Boston, Massachusetts, 1988.

10. Nicholson, Nigel, *Career Systems in Crisis: Change and Opportunity in the Information Age,* Academy of Management Executive, Vol. 10, No. 4, 1996.

11. Orfali, Robert, Harkey, Dan and Edwards, Jeri, *Essential Client/Server Survival Guide,* John Wiley & Sons, Inc., Brisbane, USA, 1994.

12. Shaffer, Steven L. and Simon, Alan R. *Transitioning to Open Systems: Concepts, Methods, ana Architecture.* Morgan Kaufman Publishers, Inc. San Francisco, California. 1995.

13. Spewak, Steven H., *Enterprise Architecture Planning,* John Wiley and Sons, Inc., New York, New York. 1995.

14. Toffler, Alvin and Heidi, Creating a New Civilization: The Politics of the Third Wave, Turner Publishing Company, Atlanta, Georgia, 1995.

15. Troy, Kathryn, *Change Management: An Overview of Current Initiatives,* The Conference Board, New York, New York, 1994.

16. Zachman, John., *A Framework for Information Systems Architecture,* IBM Systems Journal, Vol. 26, No. 3, 1987.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Director, Training and Education 1
   MCCDC, Code C46
   1019 Elliot Rd.
   Quantico, VA 22134-5027

4. Director, Marine Corps Research Center 2
   MCCDC, Code C40RC
   2040 Broadway Street
   Quantico, VA 22134-5107

5. Director, Studies and Analysis Division 1
   MCCDC, Code C45
   300 Russell Road
   Quantico, VA 22134-5130

6. Marine Corps Representative 1
   Naval Postgraduate School
   Code 037, Bld. 234, HA-220
   699 Dyer Road
   Monterey, CA 93943-5000

7. Marine Corps Tactical Systems Support Activity 1
   Technical Advisory Branch
   Attn: Major J.C. Cummiskey
   Box 555171
   Camp Pendleton, CA 92055-5080

8. Professor Magdi N. Kamel, Code SM/KA 2
   Department of Systems Management
   Naval Postgraduate School
   Monterey, CA 93943-5000

9. Professor Mark Nissen, Code SM/NI 1
Department of Systems Management
Naval Postgraduate School
Monterey, CA 93943-5000

10. The Marine Corps Institute 2
Attn: Maj. Lloyd J. Hamashin, USMC
Washington Navy Yard, 912 Poor St. S.E.,
Washington, D.C. 20391-5680

11. Maj. Clayton Evers, Jr., USMC 1
26 Brush Everard Court
Stafford, VA 22554

12. Maj. Gerald Peters, USMC 1
2 Sparrow Hill Ct
Baltimore, MD 21228